Bajtek

10

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860 – 1674

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 10(34) PAŹDZIERNIK 1988 CENA 150ZŁ

WSZYSTKO O PAMIĘCIACH

ATARI PC2

KYAN PASCAL

SM SZTANDAR MŁODYCH

NOSFERATU

KOMPUTEROWY ŚWIAT

Jakie problemy mają ludzie zajmujący się komputerami?

Tadao Takahashi z Tokio najbardziej martwi się tym, że „wirus komputerowy", będący do tej pory plagą użytkowników nowoczesnej techniki obliczeniowej w USA i kilku innych krajach Zachodu, przedostał się również na rynek japoński. Tadao Takahashi przekonał się o tym osobiście, gdy po zastosowaniu pochodzącego z USA programu komputer najpierw zaczął przekłamywać obliczenia, a następnie całkowicie „zwariował" i zaczął niszczyć kolejne wprowadzane do mego programy.

Nieuleczalna choroba komputera porażonego „wirusem" polega na tym, że samoodtwarzalny program wprowadzony kiedyś do pamięci komputera i błąkający się po niej niszczy informacje na wszystkich nośnikach. Ten, kto zaryzykuje skorzystanie z usług „zarażonego" komputera nie może liczyć na wiarygodność obliczeń oraz naraża się na kolejne straty, gdyż wszystkie wprowadzane do takiego komputera programy błyskawicznie same zaczynają być nosicielami „wirusa".

Szybkość rozprzestrzeniania się „wirusa" w krajach szeroko wykorzystujących technikę komputerową jest tak wielka, że przypomina epidemię grypy w wielotysięcznym, przepełnionym mieście. Specjaliści ustalili, że feralny program przedostał się na dyski magnetyczne z winy techników, na etapie produkcji — ale niewielka to pociecha dla pana Tadao Takahashi!

Zupełnie czym innym martwił się w połowie września 14-letni Dawid Miller uczestniczący w radziecko-brytyjskiej szkole komputerowej w Lesielidze (Gruzja) nad Morzem Czarnym. Zgłosił on do kierownictwa szkoły skargę na... zbyt obfite porcje: — Zjeść ich nie sposób — stwierdził — a wyrzucić żal, bo takie smaczne!

Dawid, podobnie jak 13 innych uczniów w Canfort, wygrał konkurs, w którym główną nagrodą był wyjazd do ZSRR. Radziecko-brytyjska szkoła komputerowa organizowana będzie w każde wakacje, na przemian w ZSRR i Wielkiej Brytanii. W ciągu roku szkolnego planuje się organizowanie telemostów oraz rozwiązywanie wybranych problemów z zakresu programowania. Szkoła ma już swój emblemat — został on wybrany spośród komputerowych projektów, wykonanych przez uczestników spotkania w Lesielidze. Główną troską Dawida jest teraz szybkie nauczenie się języka rosyjskiego — za kilka miesięcy będą przecież gościć w Canfort kolegów z ZSRR!

Natomiast pan Witold Ł. z Leszna (nazwisko i adres do wiadomości redakcji) martwi się niedoróbkami w wydaniu specjalnym „Bajtek-Commodore". „Piszę nie żeby was pognębić, tylko zmobilizować — stwierdza w liście do redakcji. — Całościowo kochani nie jesteście źli, ba, jesteście OK! ale powinniście dążyć, żeby być najlepszymi!"

Pan Witold ma pretensje przede wszystkim o to, że wydanie specjalne o Commodore „ogólnie rozczarowuje". Konkretnie rozczarowuje go „Przewodnik Amiganta" ("jeśli tak ma wyglądać — raczej zniechęci niż zachęci" — pisze nasz Czytelnik). Ma zastrzeżenia do strony graficznej niektórych publikacji ("ciekawy wywiad i artykuł zepsuty nieprzemyślanym całościowo układem"). Wytyka nam też, że „zdjęcia z ekranu publikowane na stronie 17 były już zamieszczane", choć przyznaje, że było to przy nieco innej okazji.

"Pismo musi się rozwijać, ewoluować, a nie kostnieć w konserwatyźmie złych nawyków!" — z emfazą apeluje na końcu listu Czytelnik z Leszna. Otóż w pełni się z panem zgadzamy, panie Witoldzie! Za uwagi krytyczne jesteśmy wdzięczni (pretensje do „Bajtka-Commodore" zgłosiło również kilku innych Czytelników, m.in. kolega Juliusz Zacher z Warszawy, lat 13; z satysfakcją muszę jednak dodać w tym miejscu, iż liczba opinii pozytywnych była kilkakroć większa!).

Właśnie po to, aby nasze pismo się rozwijało, opublikowaliśmy w numerze wrześniowym ankietę czytelniczą. Oddzielną ankietę zaadresowaliśmy również do odbiorców „Bajtka-Atarii 2". Chcemy aby „zwykłe", comiesięczne wydania „Bajtka" oraz wydawane przez naszą redakcję „usery" jak najpełniej zaspakajały oczekiwania Czytelników.

Jest bowiem oczywiste, że w miarę zmian na polskim rynku komputerowym, w miarę rozwijania się naszych Czytelników, również „Bajtek" musi stopniowo zmieniać swój kształt programowy i formalny. Czytelnicy zauważą to już w numerze listopadowym, który wydamy eksperymentalnie w objętości 40 kolumn, z o wiele szerszą niż dotychczas gamą tematyczną, a również — ze względu na uwarunkowania ekonomiczne — z większą liczbą reklam.

Nic mamy jeszcze w Polsce nadmiernych problemów z „wirusem" komputerowym, gdyż jest to głównie zmartwienie krajów najwyżej rozwiniętych.

Nie mamy zmartwień związanych z polsko-zagranicznymi szkołami komputerowymi, gdyż Ministerstwo Edukacji Narodowej ma na razie bardziej elementarne problemy na głowie i póki co nie jest w stanie przygotować nawet odpowiedniej liczby podręczników...

Ale dopasowanie rynku pism komputerowych do oczekiwań Czytelników leży w pełni w naszych możliwościach. Będziemy więc odpowiadając na sugestie zawarte w sympatycznym liście pana Witolda Ł. — te oczekiwania spełniać!

Waldemar Siwiński

ZA MIESIĄC:

Jeszcze 32 strony a na nich:

- ***uwaga wirusy atakują!***
- ***Monitory bez przyszłości?***
- ***Spiky Harold***
- ***Centronics do Atari 1029***
- ***Z końcówką PL oraz jak zawsze odpowiedzi na Wasze listy, gry, pożyteczne kruczki i sztuczki i inne atrakcje***

Bajtek

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"

ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych".

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SU" — kierownik zespołu „Bajtka"), Grzegorz Onichimowski (sekretarz redakcji „Bajtka"), Roman Poznański (kierownik działu klanów), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Pilaszek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski.

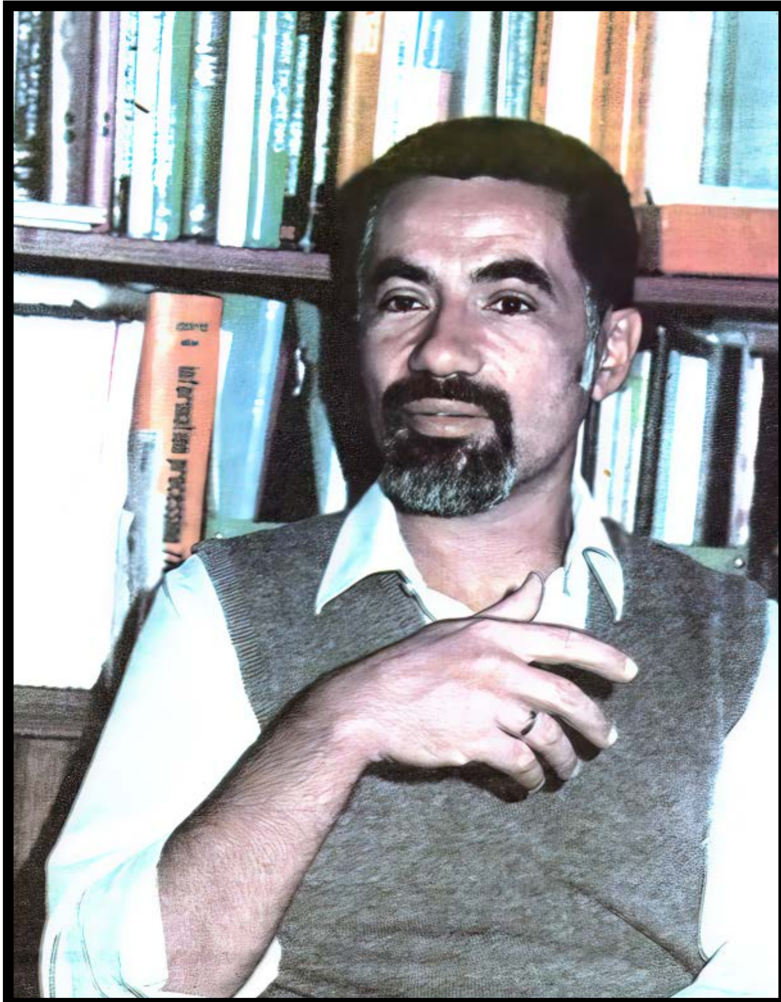
klany redagują:
Commodore — Klaudiusz Dybowski, Dominik Falkowski
Amstrad-Schneider — Jonasz Mayer
Spectrum — Marcin Przasnyski, Atari — Wojciech Zientara, Sergiusz Piotrowski.

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wółtańska

WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 150 zł.
Skład techniką CRT-200, przygotowalnią offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.
Nr zlecenia 010188 n. 150.000 egz. U-113



RZEMIEŚLNICY I NAUKOWCY



Rozmowa z doc. Janem Madeyem, dyrektorem Instytutu Informatyki Uniwersytetu Warszawskiego.

Kiedyś komputer bił po łapach. Teraz głaszcze po głowie i mówi — pomyliłeś się, zrób to jeszcze raz

— Przed czterema laty na łamach Bajtka prof. Turski wyraził pogląd, że cały ten szum, moda na mikrokomputery bardziej informatyce przeszkadza niż pomaga...

— To wszystko zależy od tego, co rozumiemy pod pojęciem informatyki. Inaczej trzeba na to patrzeć, gdy myślimy o konieczności szeroko rozumianej komputeryzacji, a inaczej jeśli chodzi o naukę. Fascynacja mikrokomputerami zamącą niekiedy obraz i wielu ludziom wydaje się, że wystarczy skończyć kurs programowania, czytać literaturę, i jeśli w dodatku potrafią napisać kilka linijek w Basicu, to są informatykami. Podobnie myli się, co znaczy komputer w szkole. Pojęcie to wcale nie jest tożsame z nauczaniem informatyki.

— A jak należałoby je rozumieć?

— Komputery stały się taką częścią cywilizacji, z którą każdy, a przede wszystkim młodzież powinna się zapoznać. Można to robić różnymi metodami odczarowywując w efekcie sprzęt do tego stopnia, by nawet ci, którzy czują do niego awersję nie uważali iż jest groźny. Trzeba uczyć posługiwania się komputerem. A przy tym odczarowywaniu wspomniany szum pomaga.

— Kiedy w takim razie szkodzi?

— Chociażby wtedy, gdy np. robione na chybcika programy naśladują nieudolnie to, co można z powodzeniem zrobić bez komputera. Programy z błędami i usterkami są antyprzykładem i mogą wręcz zniechęcić. A takich niestety jest sporo, także i na naszym rynku edukacyjnym.

Fascynacja mikrokomputerami stwarza niekiedy przekonanie, że silne komputery nie są potrzebne. Po co duży komputer, skoro np. na Spectrum można zrobić to samo?

— Jest i druga strona. Przynajmniej z początku, środowisko informatyków ustawiło się bokiem do mikrokomputerów. A przecież przełom sprzętowy, to że komputer „poszedł do ludzi", też w pewien sposób wpłynęło na metody i formy ich pracy.

— Z pewnością. Kiedyś komputer bił po łapach. Teraz głaszcze po głowie i mówi — pomyliłeś się, zrób to jeszcze raz. Grafika, dźwięk — tego w dużych systemach nie było. Ale trzeba uczyć nie informatyki, a korzystania z komputera do swoich potrzeb, swojej działalności, tak, aby stał się narzędziem pracy jak ołówek czy suwniarka.

— Wracając jeszcze do silnych komputerów. Na świecie dominują sieci komputerowe pozwalające m.in. na zdalny dostęp i swobodną wymianę informacji. U nas na razie niestety nie mamy możliwości korzystania z biblioteki np. w Stanfordzie czy innym ośrodku akademickim. Czy nasza informatyka może zatem porów-

Kiedyś komputer bił po łapach. Teraz głaszcze po głowie i mówi — pomyliłeś się, zrób to jeszcze raz nywać się ze światem?

— Jeśli chodzi o poziom nauczania — z pewnością. Polski informatyk za granicą jest ceniony. Jesteśmy poważnie traktowanym partnerem także w takich ośrodkach, gdzie sytuacja sprzętowa jest o niebo lepsza. Na przykład wielu pracowników naszego Instytutu wykłada na różnych uczelniach w Europie, USA i Kanadzie. Mamy licznych autorów podręczników wydawanych za granicą. I bardzo często są to też dyscypliny tzw. sprzętowo zależne. Czyli wydaje się, że program nauczania jaki realizowano w Instytucie był programem nowoczesnym.

— W takim razie czy mimo istnienia owej bariery sprzętowej można liczyć na to, że kiedyś staniemy się znaczącym producentem oprogramowania?

— Nie sądzę. Jest to niemożliwe nie dlatego, że brakuje nam potencjału intelektualnego. Brakuje nam nowoczesnego sprzętu, jesteśmy wciąż opóźnieni i zawsze gonimy za tym co już jest na świecie. A w związku z tym pierwszą rzeczą jaką robimy jest dostosowywanie zachodnich programów narzędziowych do naszych warunków. Swoich natomiast nie tworzymy, bo istnieją już zwykle dobre i sprawdzone „importowane".

— A np. kompilatory języków?

— Tutaj rzeczywiście mamy pewne tradycje. Spośród krajów RWPG Polska była pierwszym krajem gdzie zauważono język Prolog. W naszym Instytucie powstał język Loglan, powoli zyskujący zwolenników w wielu krajach. Są też inne możliwości. Jeżeli np. w analizie numerycznej zostanie wymyślona nowa metoda rozwiązywania równań różniczkowych to można zrobić oprogramowanie. Tutaj możemy być konkurencyjni. Ale główny rynek zbytu, na którym sprzedaje się w milionach egzemplarzy, gdzie robi się biznes jest poza naszym zasięgiem. Także z tego powodu, że większość narzędzi za pomocą których jest tworzone oprogramowanie jest po prostu kradziona.

— Ponoć nie stać nas na ich kupowanie.

— Argumenty tego rodzaju nie przekonują mnie. To jest tak, jak gdyby kraść w supermarkecie towary tylko dlatego, że u nas takich nie ma. W dodatku towary kosztujące grosze, równowartość pary butów, czy książki. Tym samym kopiemy pod sobą dołek. Nasi informatycy zaczynają mieć problemy z wizami np. do USA. Poza tym jak w takiej sytuacji można mówić o edukacji młodych ludzi skoro od samego początku uczą się, że warto jest kraść? Jest to z pewnością trudny problem, ale wymaga jak najszybszych rozwiązań, gdyż przynosi jak na razie o wiele więcej strat niż korzyści.

— W wyścigu ze światem nie mamy zatem szans. A w kraju? Jak u nas wygląda problem konkurencyjności?

— Jeśli chodzi o konkurencyjność w sensie produkowania chociażby programów aplikacyjnych np. typu sterowania procesem chemicznym to jest to czysta abstrakcja. Z kim mamy konkurować? Dla kogo to robić? Jeżeli już, to chyba wyłącznie dla siebie.

— Na świecie prowadzi się optymalizację np. produkcji amoniaku biorąc pod uwagę wydajność, ochronę środowiska, problemy magazynowania, u nas problem sprowadza się z reguły do tego, aby produkować najwięcej. A w takim wypadku nie potrzeba wyrafinowanych systemów.

— Wróćmy jednak do samej informatyki. Fakt, iż komputery zeszły na niższy poziom powoduje, że trzeba zmieniać styl i formy nauczania.

— A także zróżnicować szkolenie informatyczne. W wielu miejscach pracy wcale nie muszą pracować ludzie z wykształceniem uniwersyteckim. To jest kolejne nieporozumienie. Natomiast tam, gdzie robi się rozwiązania specjalistyczne czy też w miejscach gdzie rozwija się informatykę, wykształcenie powinno być na wysokim poziomie.

— Kogo w takim razie można nazwać informatykiem? Absolwenta szkoły pomaturalnej, zawodowej, uniwersytetu?

— To zależy od potrzeb. W wielu zakładach wystarczy, aby był dobrym rzemieślnikiem, a nie naukowcem.

— Od pewnego czasu np. w oświacie dominuje coś, co można by nazwać pewnego rodzaju zadowoleniem z tego, co już zrobiono. Jak ocenia pan to co się dzieje w szkołach?

— Jest to dla mnie, jako dla przewodniczącego zespołu doradczego byłego ministra oświaty i wychowania sprawa bardzo bolesna. Do dzisiaj nie uzyskałem odpowiedzi nowego ministra, co do dalszych losów zespołu. Jak na razie jedno jest wiadome. Nie udało się nam uzyskać tego, o co walczyliśmy za poprzednich ministrów, aby w nowych strukturach była wydzielona jednostka organizacyjna zajmująca się całokształtem spraw związanych z wprowadzaniem komputerów do szkół. A jest ich bardzo dużo, od sprzętu począwszy, na kształceniu nauczycieli skończywszy. Są oczywiście różnego rodzaju sensowne działania, np. powołano m.in. wizytatorów-informatyków, z mniejszymi lub większymi kłopotami wprowadza się „Juniora" do szkół. Brak jednak klarownych rozwiązań strukturalnych. A bez tego o postępie trudno mówić.

— Czy ma pan jakieś specjalne życzenia, dotyczące absolwentów szkół średnich — pańskich przyszłych studentów?

— Trudno powiedzieć. Jeśli dana osoba miałaby być przyszłym informatykiem-naukowcem to życzyłbym sobie, aby na przedmiocie elementy informatyki była uczona właściwie. Właściwie, to znaczy tak, aby później nie trzeba było oduczać ją złych nawyków, które bardzo ciężko wyplenić. Nie mam specjalnych wymagań. Może oprócz jednego, aby absolwent chciał pracować twórczo. Ideałem byłoby, gdyby tacy ludzie trafiali na studia.

Rozmawiali:
Krzysztof Małecki
Grzegorz Onichimowski

ZAGRANICZNE KOMPUTERY DLA FRANÇUCKICH LICEÓW

Francuski przemysł komputerowy, mający i tak trudności na rynku, przegrał podczas ostatnich wakacji „bitwę o szkoły". W połowie sierpnia nowy minister oświaty tego kraju Lionel Jospin podjął decyzję, że zdecydowana większość komputerów osobistych, które staną jeszcze w tym roku w pracowniach informatycznych liceów francuskich pochodzić będzie z zagranicy. Cios dla przemysłu francuskiego tym cięższy, że nowy, socjalistyczny rząd zdecydował się równocześnie zwiększyć znacznie skalę zamówienia. Nowych komputerów PC ma być we francuskich szkołach średnich aż 13 120, co wraz z otrzymanymi wcześniej gwarantuje dość swobodny dostęp każdemu z uczniów. Rząd zgodził się zapłacić za nowe zakupy aż pięciokrotnie więcej niż początkowo planowano — 156 min franków (ok. 25 mln. dolarów).
Gdy przed trzema laty poprzedni rząd socjalistyczny inicjował w szkołach akcję „informatyka dla wszystkich", uważano, że stanie się ona równocześnie wsparciem dla słabnącego rodzimego przemysłu. Wówczas zdecydowano się zakupić do szkół przestarzały już nieco wtedy typ T-07 francuskiej firmy Thomson, odrzucając ofertę amerykańskiego Mcintosha w postaci modelu „Apple", za którym opowiadała się zdecydowana większość specjalistów. Dla Thomsona była to gratka nie lada, bowiem do szkół trafiło wtedy aż 100 tysięcy komputerów tej firmy.
Obecnie narodowe sentymenty zostawiono na boku, kierując się wyłącznie zasadą: decyduje cena, wartość techniczna i ich wzajemny stosunek. W wyniku przetargu prawie 6 tysięcy komputerów dostarczy amerykańska firma Victor należąca do szwedzkiej grupy Dataronic, 2370 włoska Olivetti-Legobax, a resztę małe francuskie wytwórnie SMT (3470), Leanord (664), Buli (525) i Forum (184). Tak więc ponad 60 procent zamówień dostała zagranica, a w przypadku drukarek nawet całe 100 procent, bowiem wszystkie, czyli 5056 sztuk dostarczą japońskie firmy NEC i Epson.
Tym razem Thomson nie otrzymał ani jednego zamówienia. Ta znana francuska firma elektroniczna ponosiła w ostatnich latach na rynku komputerowym same porażki. W 1986 z planowanych 300 tys. sztuk sprzedano tylko 100 tys. Rok później planowano „kilkadziesiąt tysięcy", a sprzedano zaledwie 10 tys. Po ostatnim zerowym interesie z ministerstwem oświaty mówi się, że Thomson w ogóle zrezygnuje z produkcji komputerów osobistych.
Również i inne francuskie firmy skłaniają się ku podobnym wnioskom. Przemysł francuski dostarcza zaledwie 15 procent komputerów sprzedawanych we Francji i zaledwie 5 procent w całej Europie Zachodniej. Firma SMT-Goupil narzeka na nierówną konkurencję — komputery sprowadzone w całości w Tajwanu obciąża się jedynie 4-procentowym podatkiem, składane we Francji z tajwańskich części — 15 procentowym. Firma Buli, której udział na rynku francuskim spadł ostatnio z 12,5 do 7,4 procenta, zastanawia się nad zmianą strategii, podobnie jak i mniejsi producenci, produkujący jeszcze drożej, ze względu na krótsze serie

(jw)

POROZUMIENIE
IBM–AMSTRAD

Ogłoszone w lipcu porozumienie między firmami Amstrad i IBM, przewidujące wzajemne wykorzystywanie patentów wywołało duże zaskoczenie w międzynarodowej branży komputerowej. Bardzo trudno byłoby znaleźć przykład bardziej odmiennych od siebie przedsiębiorstw — napisał londyński „Times" komentując to wydarzenie. Twórca Amstrada, Alan Sugar, znany jest ze zdobywania nowych rynków poprzez śmiałe obniżki cen swych wyrobów — w tym również wielu typów komputerów osobistych wzorowanych, a zdaniem niektórych wprost kopiowanych z IBM, i sprzedawanych następnie o połowę taniej. Ku zaskoczeniu wielu obecnie zawarł on ze swym potężnym konkurentem porozumienie pozwalające oficjalnie wytwarzać kopie komputerów osobistych IBM — giganta, który bardzo niechętnie godzi się z faktem, że rynkiem zarządzają również prawa cenowe, a nie tylko uznanie dla znakomitej jakości.

Oświadczenie opublikowane z tej okazji przez Amstrada stwierdza, że porozumienie udostępnia tej firmie ogólnoświatową licencję na wszystkie patenty IBM, włącznie z tymi które odnoszą się do komputerów osobistych IBM, także tych, które należą do najnowszej serii PS2. Jeśliby to ostatnie okazało się w pełni prawdą, to znaczy jeśli Alan Sugar otrzymałby zezwolenie na sprzedaż licencjowanych — i niewątpliwie tańszych kopii systemu PS2, byłoby to prawdziwą niespodzianką, jako że sama IBM nie zdołała jeszcze dobrze usadowić się z tymi wyrobami na rynkach międzynarodowych.

Na to pytanie padła już jednak częściowa odpowiedź. Przedstawiciele IBM całkowicie odrzucili przypuszczenia, że układ pozwala również na kopiowanie architektury mikrokanałowej (MCA-MICRO Chanel Architecture), stanowiącej klucz od wytwarzania PS 2. Układ odnosi się jedynie do rozwiązań technicznych, a każdy, kto chciałby kopiować PS 2 musi być bardzo ostrożny, by nie naruszać praw autorskich dotyczących MCA. — Stwierdzał rzecznik IBM.

Jeśli chodzi natomiast o standardowe rozwiązania IBM-owskiej serii PC, to obecnie nie ma już żadnych wątpliwości co do tego, że Alan Subar będzie mógł w pełni legalnie zająć się produkowaniem licencjonowanych już kopii jej modeli. Przewiduje się, że już wczesną jesienią Amstrad wypuści na rynek własne wersje silniejszych komputerów, opartych na chipach 286 i 386.

Nowy układ niewątpliwie znacznie poprawi reputację Amstrada tam, gdzie krytykowano go za plagiaty. Z zadowoleniem ulgą powitają go również zapewne inne firmy produkujące klony maszyn IBM. IBM z kolei będzie dążył — jak zwykle czyni to w takich przypadkach — do wyciągnięcia jak największych korzyści z opłat licencyjnych, które niewątpliwie sięgać będą wielu milionów dolarów. Być może amerykański gigant przeznaczy te sumy na zakup akcji Amstrada, które jedynie w ciągu ostatniego roku podwoiły swą wartość na giełdach, a po ogłoszeniu układu od razu poszły w górę o dalsze 3 procent.

(jw)

CRAY OTRZYMAŁ ZAMÓWIENIA NA JESZCZE NIE ZBUDOWANY SUPERKOMPUTER

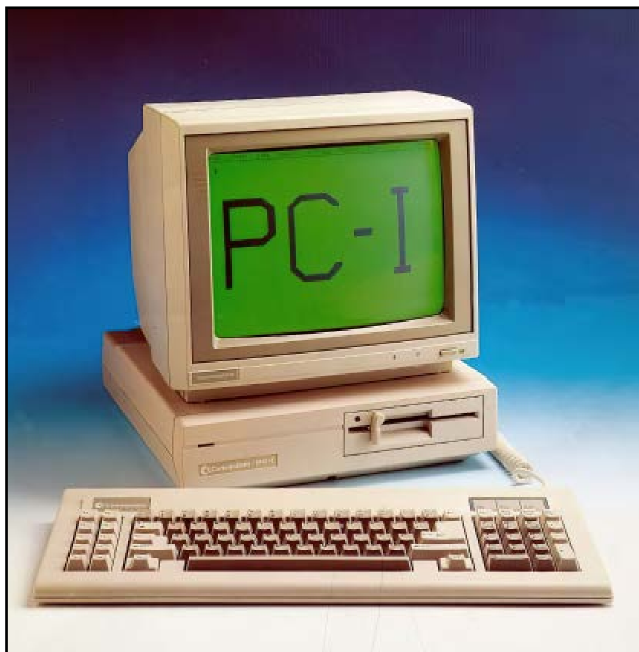
Największa na świecie firma produkująca superkomputery — Cray Research Inc. już w tej chwili ma przyszłościowy model najszybszego na świecie superkomputera, który — nota bene — nie istnieje jeszcze nawet w formie prototypu. Mająca kosztować 25 mln dolarów maszyna została zamówiona z góry, na dwa lata przed spodziewaną dostawą, przez Pittsburg Supercomputing Center, będący wspólnym przedsiębiorstwem Uniwersytetu Pittsburskiego, Uniwersytetu Carnegie-Mellon i firmy Westinghouse.

Przedmiotem transakcji ogłoszonej w połowie sierpnia jest model Cray 3, który według przewidywań będzie najpotężniejszym superkomputerem na świecie, zdolnym do przeprowadzania działań z szybkością co najmniej trzykrotnie większą niż obecne najszybsze urządzenia. Szef firmy, jej twórca i główny specjalista Seymour Cray ma nadzieję, że jej prototyp będzie gotów do prób przed upływem tego roku. Przedstawiciele firmy podali, że jest to pierwszy przypadek zawarcia kontraktu w tak wczesnym stadium opracowania nowej jednostki i podkreślili, że projekt wymaga jeszcze wielkiego nakładu pracy. Nie ma żadnych powodów, by sądzić, że dzieło to nie zostanie uwieńczone sukcesem — powiedział wiceprezes firmy Edward A. Mast — choć w przypadkach całkiem nowych wyrobów nigdy nie ma się całkowitej pewności, aż do ostatniej chwili.

Jeśli Cray spełni wszystkie oczekiwania, będzie to już trzeci kolejny model Seymoura Craya wyprzedzajay wszystko, co w tej dziedzinie zrobiono dotąd na świecie, pod względem siły i szybkości komputera.

(j.w)

PC — ofertą dla polskiej rodziny



Do takiego wniosku doszło kierownictwo „Pewexu"decydując się na wprowadzenie do swoich sklepów obok komputerów ośmiobitowych także pracujący w standardzie MS DOS Commodore PC 1.

PC 1 ma, w zamierzeniu swoich konstruktorów, być pomostem pomiędzy komputerami domowymi i osobistymi. W pełni zgodny ze standardem PC XT jest jednak wykonany w znacznie nowocześniejszej technologii niż większość dalekowschodnich „klonów". Technologia ta pozwoliła na zmieszczenie na płycie głównej 512 KB RAM kontrolera dysków elastycznych, adaptera typu CGA oraz łączy: równoległego i szeregowego. PC 1 wykorzystuje standardowo system operacyjny MS-DOS 3.2 i monitor monochromatyczny (można podłączyć również kolorowy). Najprawdopodobniej jednak w ofercie „Pewexu" znajdzie się PC 1 wraz z monitorem monochromatycznym.

Niestety, jak dotąd, nie znamy ilości zakupionych do Polski komputerów PC 1, ani ich ceny detalicznej. Jeśli jednak nie będzie ona zbyt wygórowana może się on okazać wspaniałym narzędziem dla wszystkich tych, którzy chcieliby także w domu dysponować maszyną klasy PC. Patrząc na zdjęcie można dojść do wniosku, że jest to z pewnością jeden z najbardziej elegancko prezentujących się odmian PC.

(go)



NA POGRANICZU DWÓCH SYSTEMÓW

Pierwszym najpopularniejszym systemem operacyjnym, który opanował świat komputerów 8-bitowych był CP/M. Jego najnowsza wersja — CP/M Plus znana jest użytkownikom małych Amstradów i Commodore 128. Wielu posiadaczy tych komputerów ma szanse w miejscu swojej pracy korzystać z komputerów 16-bitowych typu IBM PC. Przejście z CP/M'u do MS-DOS'a, jest dość naturalne. Jeśli w pracy napiszemy na IBM'ie DIR to dostaniemy podobną informację do tej którą otrzymamy po wystukaniu tej sekwencji na klawiaturze naszego Amstrada. To samo dotyczy rozkazów ERASE, TYPE, PIP (COPY) i innych. Idea działania obu systemów jest taka sama.

Bardzo ciekawą konstrukcją stojącą na pograniczu CP/M i MS-DOS jest przenośny, bezdyskowy komputer firmy Husky. Model Hawk 8/16 wyposażony w dwa procesory, to dwie maszyny w jednej. Pracuje on w dwóch systemach operacyjnych: MS-DOS 3.21 i DEMOS (odpowiednik CP/M 2.2). Z powodu braku stacji dysków i wyposażenia w ciekłokrystaliczny ekran jest bardzo lekki, waży poniżej kilograma Pamięć RAM typu CMOS podtrzymywana jest bateryjnie po odłączeniu głównego zasilania, a zbiory zawarte w niej mogą być przesłane w każdym z działających systemów przy użyciu dwóch złączy typu RS-232. Duża część niezbędnego oprogramowania aplikacyjnego dostępna jest w pamięci ROM.

Abstrahując od zastosowanego systemu operacyjnego jest to sprzęt zbliżony w swej filozofii do ostatniej nowości Sir Clive'a Sinclaire'a jaką jest Z88. Komputery tego typu doskonale nadają się do prac w terenie. Zbieranie danych pomiarowych, statystycznych, czy pisanie tekstów w podróży to niektóre z typowych zastosowań.

Zniechęca trochę wysoka cena za to cacko technologii — ponad 1300 funtów.

Parametry techniczne komputera Husky Hawk 8/16:
Główny procesor: 80C88, 6.144 Mhz
RAM: 640 kB CMOS
ROM: 128 kB CMOS
Dodatkowy procesor: HD64B180 (zgodny z Z80), 6.144 Mhz
RAM: 352 kB CMOS, w tym:
— 260 kB dysk
— 64 kB TPA
— 4 kB pamięć ekranu
ROM: 48 kB CMOS, w tym:
— DEMOS
— Locomotive Basic
Pamięć zewnętrzna: brak
Ekran: wyświetlacz LCD 240 x 64 punktów
Klawiatura: uproszczona wersja klawiatury IBM PC
Złącza: szeregowe RS 232 (25) do komunikacji i RS 232 (8) do drukarki
Główny system op.: MS-DOS 3.21
drugi system op.: DEMOS (and. Disk EMulating Operating System)

(j.m.)



SUN 4/110 WORKSTATION

Słowo Workstation nie znalazło jeszcze dobrego tłumaczenia w naszym języku, ale zdążyło się już zadomowić wśród osób zainteresowanych czymś więcej, niż typowe komputery.

Zagadnienia projektowania wspomaganego komputerowo i ogólnie rozumianej sztucznej inteligencji są nieslychanie żarłoczne jeśli chodzi o pamięć i czas procesora. Dawniej tylko duże komputery były w stanie częściowo sprostać tym zagadnieniom. Obecnie idzie im to trochę łatwiej, a jednocześnie postęp technologii pozwala część tych zadań przenieść na mniejsze maszyny o wielokrotnie niższej cenie, dostępnej dla typowych biur konstrukcyjnych czy wydziałów wyższych uczelni. Takie mikrokomputery jak DeskPro 386/20 czy Macintosh II zbliżają się do granic zarezerwowanych dla maszyn typu Workstation. Niestety jest to dopiero przedsmak tego co możemy mieć jeśli kupimy sprzęt o nazwie SUN 4/110.

Zacznijmy od procesora w jaki wyposażona jest ta maszyna. 32-bitowy SPARC (ang. Scalable Procesor ARChitecture) osiąga olbrzymią moc obliczeniową rzędu 7 MIPS (miliona instrukcji na sekundę) i w przypadku zastosowania koprocesora numerycznego jest pięciokrotnie szybszy od VAX'a 11/780.

Procesor ten zrealizowany w architekturze typu RISC (ang. Reduced Instruction Set Computer) dysponuje 50 rozkazami, z których każdy jest wykonywany w ciągu jednego cyklu zegara, 120 rejestrów zorganizowanych w 5 „okien" po 24 rejestry każde, pozwala na bardzo efektywną obsługę odwołań do procedur, generowanych przez kompilatory języków wysokiego poziomu takich jak C lub Pascal.

Kolejnym wyróżnikiem jest zdolność tego komputera do przetwarzania obrazów. Specjalnie zaprojektowany układ MMU (Memory Managment Unit) pozwala na zaadresowanie 256 MB pamięci ekranu. Rozdzielczość obrazu w 256 kolorach wynosi 1600 na 1280 punktów. Zastosowanie specjalizowanego video koprocesora GP2 umożliwia narysowanie 150 000 3- wymiarowych wektorów w ciągu 1 sekundy. Pamięć operacyjna Workstation SUN 4/110 w typowej konfiguracji wynosi 8 MB. Możliwe jest jej rozszerzenie do 32 MB. Charakterystyczna jest również duża pojemność twardego dysku, standardowego—327 MB.

Systemem operacyjnym stosowanym w przypadku Workstation jest najczęściej UNIX. I w tym wypadku SUN nie odbiega od normy. Ułatwieniem w pracy są oferowane nakładki: SunView, X-Windows i NeWs. Oprócz języków wysokiego poziomu takich jak: C, Pascal, Lisp, Modula-2, VMS Fortran dostępne są kompilatory skrośne: SPARC-to-680x0 i 680x0-to-SPARC. Typową cechą Workstation jest też możliwość łączenia ich w sieć z innymi większymi komputerami.

Dla użytkownika korzystanie z tego typu sprzętu nie jest trudniejsze niż praca na mikrokomputerze osobistym. Co najwyżej zauważy on kilkakrotnie większą pamięć operacyjną, szybkość działania, rozdzielczość ekranu, czy pojemność twardego dysku. Właśnie te zmiany decydują o przejściu do nowej klasy sprzętu pozwalającego na realizację zadań wykonywanych do tej pory na bardzo dużych komputerach.

(jm)

XT trzyma się mocno

Komputery standardu IBM PC zadomowiły się w naszym kraju na dobre. Kto ma pieniądze może przebierać między różnymi odmianami tego sprzętu. Za bardzo dużą sumę możemy kupić systemy 32-bitowe z możliwością pracy w systemie OS/2. Za trochę mniejszą staniemy się użytkownikami komputerów AT. Najmniej musimy zapłacić za „stare, poczciwe" XT, uważane przez niektórych posiadaczy AT za obiekt historyczny. Tymczasem jest to sprzęt, którego najnowsze konstrukcje śmiało mogą konkurować ze swoimi następcami serii AT.

Podstawowe różnice między modelami XT i AT można zebrać w poniższej tabelce: XT. AT.

1. Płyta główna 640kB RAM, zegar 4,77/8 Mhz	1. Płyta główia 1MB RAM. zegar 6/8 Mhz
2. dwa dyski miękkie po 360 kB	2. dyski miękkie: 1,2MB i 360 kB
3 dysk twardy 20MB	3. dysk twardy 30 i więcej MB
4 brak rozszerzeń pamięci RAM	4. rozszerzenie pamięci RAM do 4MB

Obecnie najnowsze komputery typu XT wyposażone są w główną płytę zawierającą 1MB pamięci. (640 kB widziane przez DOS i 384 kB bardzo użytecznego ramdysku). Częstość zegara może być zwiększona do 12—14 Mhz, ale nowym standardem jest 10 Mhz.

Dzięki specjalnemu kontrolerowi możliwe jest dołączenie dysków 1,2 MB i to nawet dwóch. Zmiana techniki zapisu twardego dysku pozwala uzyskać pojemność 32 MB. Dostępne są także rozszerzenia pamięci 2MB, pozwalające na założenie w najprostszym przypadku dużego RAM-dysku, znacznie ułatwiającego pracę.

Jest jeszcze jeden argument przemawiający za XT. W językach wysokiego poziomu specjalnie generujących kod wynikowy, np. kompilator FORTRAN-u PROFORT, programy współpracujące z koprocesorem działają ponad 50 proc.!!! szybciej.

Oczywiście nie należy zapominać o zmianach na lepsze w technologii AT, ale jeśli zależy nam na sprzęcie, który ma być intensywnie wykorzystywany do obliczeń numerycznych, to za 50-75 proc. ceny AT dostaniemy efektywniejsze w tej dziedzinie XT. Do bardzo szybkich AT (16Mhz) nie ma niestety równie szybkich koprocesorów.

Komputer XT z zegarem 10Mhz i odpowiednim koprocesorem (8087-1) osiąga 20 proc. wydajności dużego komputera CDC 6000, produkowanego na przełomie lat sześćdziesiątych i siedemdziesiątych przez amerykańską firmę Control Data Corporation. Przez wiele lat była to najszybsza, dostępna w Polsce maszyna, na której wychowało się kilka pokoleń informatyków.

Przy okazji ciekawe porównanie między CDC a IBM PC, ilustrujące tezę, że łatwiej jest zbudować dobry technicznie komputer niż stworzyć równie dobre oprogramowanie do niego. Efektywność generacji kodu wynikowego dla typowego kompilatora FORTRAN-u na CDC sięga ponad 95 proc, tzn, dobry programista posługujący się asemblerem jest w stanie poprawić szybkość programu tylko o kilka procent. Dla wspomnianego wcześniej PROFORT-a, działającego na IBM'ie podobny współczynnik jest rzędu 30-40 proc. Oznacza to, że musimy poczekać jeszcze kilka lat na jakąś wersję super C lub super Pascala, która pozwoli liczyć na naszym biurowym IBM-ie tak szybko jak na dużej maszynie.

Jeśli zależy nam na czasie, a pieniądze nie są dla nas lub dla naszej instytucji problemem, możemy pozwolić sobie na zakup sprzętu klasy SUN 4/110. Za kilkanaście tysięcy dolarów otrzymamy komputer o niewyobrażalnej, jak na nasze krajowe warunki, mocy obliczeniowej pięciu minikomputerów VAX 11/78.

(jm)

PRZENOŚNY MACINTOSH



Macintosh, jeden z najpopularniejszych w USA mikrokomputerów doczekał się obecnie swojej wersji przenośnej. Nosi ona nazwę Dynamac EL a została wyprodukowana przez współpracującą z Apple firmę Dynamac Computer Products.

Dynamac EL wykorzystuje układy elektroniczne z modelu Mac Plus. W ich skład wchodzi min. procesor 68000 pracujący z częstotliwością 7.83 MHz, ostatnią wersją ROM o pojemności 128 KB, 1 MB RAM (rozszerzalne do 2.5 lub 4 MB) i stacją dyskietek 3.5 calowych. Na tym jednak kończy się rejestr urządzeń rodem bezpośrednio z Macintosha. Dynamac posiada zaprojektowane specjalnie układy pozwalające na dołączenie zewnętrznej klawiatury, dysku twardego i wewnętrznego modemu. Niezwykle nowatorskie rozwiązania techniczne zawiera elektroluminescencyjny ekran komputera o żółtym lub pomarańczowym tle. W opinii ekspertów miesięcznika „Byte" jest on lepszy od standardowego ekranu „stacjonarnego" Maca. Jego rozdzielczość — 640x400 punktów przy wymiarach 12 na 19 centymetrów jest rzeczywiście imponująca jak na model przenośny.

Dynamac EL akceptuje w pełni oprogramowanie Macintoshna w swoim bogactwie porównywalne chyba tylko do przeznaczonego na komputery PC. Do Polski jednak prawdopodobnie nie trafi nie tylko z powodu małej popularności na naszym rynku wyrobów ze znakiem jabłuszka. Możliwość posługiwania się tym komputerem w podróży czy na urlopie nie równoważą chyba ceny, która w przypadku podstawowej konfiguracji sięga... 5 tys. dolarów.

(go)

80 znaków w wierszu

Profesjonalne komputery umożliwiają zwykle uzyskanie na ekranie monitora 80 znaków w każdej linii. W porównaniu z tym, 40 znaków w linii dostępne na Atari, to śmiesznie mało. Zamieszczamy więc program, który umożliwia podwojenie tej liczby.

Program działa w trybie graficznym 8 i składa się z dwóch części. Pierwsza generuje nowy zestaw znaków, który umieszczony jest od adresu 31744 i zajmuje 1 KB pamięci (z tego względu program, który będzie wykorzystywał przedstawioną procedurę musi zostawiać w grafice 8 przynajmniej 1,5 KB wolnej pamięci). Właściwa procedura stanowi drugą część programu. Wczytywana jest ona na 6 stronę pamięci. Po wczytaniu obu części program w Basicu może być usunięty (w tym celu należy zamienić instrukcję END w linii 1050 na NEW lub RUN "C:").

Aby uzyskać tekst trzeba teraz włączyć tryb GRAPHICS 8, a następnie użyć instrukcji:

I = USR(1536, ADR(T$),X, Y,LEN(T$))

gdzie T$ jest tekstem, który chcemy wyświetlić: X jest współrzędną poziomą początku tekstu (od 0 do 79, a więc z dokładnością 4 punktów ekranu); zaś Y jest współrzędną pionową (od 0 do 184, a z oknem tekstowym do 152),

Umieszczony na ekranie tekst wymazuje tło, na którym się pojawi (oprócz pierwszego znaku przy nieparzystej wartości X). Gdy zamierzamy wykonać nadruk na istniejącym rysunku, to należy przed wywołaniem procedury przez USR, wpisać instrukcję POKE 1770,128. Teraz tekst w miejscach mających kolor 1 będzie miał kolor 0 i odwrotnie. POKE 1664,13 spowoduje, że tekst zawsze będzie miał kolor 1. Normalny tryb pracy programu przywracają instrukcje POKE 1770,131 i POKE 1664,77.

Przedstawiona procedura ma jednak kilka ograniczeń. Przede wszystkim nie może wykonywać napisów w negatywie (inverse video). Nie pozwala także na zmianę miejsca umieszczenia pamięci obrazu. To ostatnie ograniczenie można ominąć wpisując instrukcje POKE 1615,H i POKE 1622,L, gdzie H i L to odpowiednio starszy i młodszy bajt adresu pamięci obrazu. Ponieważ przesunięcie obrazu może spowodować zniszczenie obszaru zajętego przez zestaw znaków, to również można go przenieść w inne miejsce. W tym celu trzeba zmienić wartość 31744 w linii 10 na nową i wpisać starszy bajt nowego adresu przez POKE 1577,A/256. Trzeba przy tym pamiętać, aby zestaw znaków rozpoczynał się od początku strony, a więc od adresu podzielnego przez 256.

Efektywne wykorzystanie programu wymaga dołączenia go do innego programu w Basicu lub (po disasemblacji) do programu w kodzie maszynowym. Można także spowodować, aby program po umieszczeniu w pamięci zestawu znaków i procedury maszynowej uruchamiał inny program w Basicu.

Michał Skowroński

```
CN  0 REM *** 80 ZNAKOW W LINII ***
NK  1 REM *** Michal Skowronski ***
GK  2 REM ****** POZNAN 88-03 *****
VI  10 GRAPHICS 0:A=31744
JI  20 FOR I=1 TO 128:SUMA=0:TRAP 70
SO  30 FOR J=1 TO 8:READ X:POKE A,X
VQ  40 SUMA=SUMA+X:A=A+1:NEXT J
XH  50 READ SPR:IF SPR<>SUMA THEN 70
SL  60 NEXT I:GOTO 1000
OR  70 ? "BLAD DANYCH W LINII ";I*5+95
UF  80 LIST I*5+95:? "DANA NR. ";J
YX  90 END
JX  100 DATA 0,5,7,7,7,2,2,0,30
AS  105 DATA 6,6,6,7,7,6,6,6,50
DX  110 DATA 1,1,1,1,1,1,1,1,8
PH  115 DATA 6,6,6,14,14,0,0,0,46
FI  120 DATA 6,6,6,14,14,6,6,6,64
XR  125 DATA 0,0,0,14,14,6,6,6,46
OV  130 DATA 1,1,2,2,4,4,8,8,30
II  135 DATA 8,8,4,4,2,2,1,1,30
WX  140 DATA 1,1,3,3,7,7,15,15,52
CC  145 DATA 0,0,0,0,3,3,3,3,12
MV  150 DATA 8,8,12,12,14,14,15,15,98
YM  155 DATA 3,3,3,3,0,0,0,0,12
PX  160 DATA 12,12,12,12,0,0,0,0,48
WW  165 DATA 15,0,0,0,0,0,0,0,15
AR  170 DATA 0,0,0,0,0,0,0,15,15
WS  175 DATA 0,0,0,0,12,12,12,12,48
KE  180 DATA 0,2,2,5,5,2,7,0,23
TC  185 DATA 0,0,0,7,7,6,6,6,32
AR  190 DATA 0,0,0,15,15,0,0,0,30
KC  195 DATA 6,6,6,15,15,6,6,6,66
YW  200 DATA 0,6,15,15,15,6,0,0,57
ZV  205 DATA 0,0,0,0,15,15,15,15,60
OO  210 DATA 8,8,8,8,8,8,8,8,64
BH  215 DATA 0,0,0,15,15,6,6,6,48
SM  220 DATA 6,6,6,15,15,0,0,0,48
UH  225 DATA 12,12,12,12,12,12,12,12,96
LG  230 DATA 6,6,6,7,7,0,0,0,32
ND  235 DATA 6,4,6,4,7,2,3,0,32
HP  240 DATA 0,2,7,2,2,2,2,0,17
JI  245 DATA 0,2,2,2,2,7,2,0,17
DH  250 DATA 0,0,0,4,15,4,0,0,23
GL  255 DATA 0,0,0,2,15,2,0,0,19
RC  260 DATA 0,0,0,0,0,0,0,0,0
ZR  265 DATA 2,2,2,2,2,0,2,0,12
UG  270 DATA 5,5,0,0,0,0,0,0,10
YX  275 DATA 0,0,6,15,6,15,6,0,48
DQ  280 DATA 2,3,4,2,1,6,2,0,20
MU  285 DATA 0,5,1,2,2,4,5,0,19
NA  290 DATA 0,4,10,4,11,10,5,0,44
XP  295 DATA 2,2,0,0,0,0,0,0,4
KJ  300 DATA 2,2,4,4,4,4,2,2,24
KY  305 DATA 4,4,2,2,2,2,4,4,24
HB  310 DATA 0,2,7,2,2,7,2,0,22
XZ  315 DATA 0,0,2,7,2,0,0,0,11
CF  320 DATA 0,0,0,0,0,0,2,4,6
CS  325 DATA 0,0,0,7,0,0,0,0,7
YH  330 DATA 0,0,0,0,0,0,4,0,4
HO  335 DATA 1,1,2,2,2,4,4,0,16
SA  340 DATA 2,5,5,5,5,5,2,0,29
AX  345 DATA 1,3,5,1,1,1,1,0,13
JG  350 DATA 2,5,1,2,2,4,7,0,23
JW  355 DATA 2,5,1,2,1,5,2,0,18
IC  360 DATA 1,3,5,5,7,1,1,0,23
MK  365 DATA 7,4,6,1,1,5,2,0,26
SI  370 DATA 2,5,4,6,5,5,2,0,29
KM  375 DATA 7,1,2,2,4,4,4,0,24
NJ  380 DATA 2,5,5,2,5,5,2,0,26
IR  385 DATA 2,5,5,3,1,5,2,0,23
XZ  390 DATA 0,0,2,0,0,2,0,0,4
FY  395 DATA 0,0,2,0,0,2,4,0,8
VS  400 DATA 0,1,2,4,2,1,0,0,10
DL  405 DATA 0,0,7,0,0,7,0,0,14
AT  410 DATA 0,4,2,1,2,4,0,0,13
CH  415 DATA 2,5,1,2,2,0,2,0,14
RT  420 DATA 0,6,9,11,11,8,6,0,51
SB  425 DATA 2,5,5,7,5,5,5,0,34
XM  430 DATA 6,5,5,6,5,5,6,0,38
NP  435 DATA 2,5,4,4,4,5,2,0,26
VX  440 DATA 6,5,5,5,5,5,6,0,37
UV  445 DATA 7,4,4,6,4,4,7,0,36
OR  450 DATA 7,4,4,6,4,4,4,0,33
RJ  455 DATA 2,5,4,4,5,5,3,0,28
WB  460 DATA 5,5,5,7,5,5,5,0,37
KN  465 DATA 7,2,2,2,2,2,7,0,24
IY  470 DATA 7,1,1,1,1,5,2,0,18
WQ  475 DATA 5,5,6,6,5,5,5,0,37
MZ  480 DATA 4,4,4,4,4,4,7,0,31
ZQ  485 DATA 5,7,7,5,5,5,5,0,39
ZL  490 DATA 5,5,7,7,5,5,5,0,39
TA  495 DATA 2,5,5,5,5,5,2,0,29
PZ  500 DATA 6,5,5,6,4,4,4,0,34
KY  505 DATA 2,5,5,5,5,5,2,1,30
VM  510 DATA 6,5,5,6,5,5,5,0,37
EZ  515 DATA 2,5,4,2,1,5,2,0,21
JQ  520 DATA 7,2,2,2,2,2,2,0,19
NK  525 DATA 5,5,5,5,5,5,2,0,32
QW  530 DATA 5,5,5,5,5,2,2,0,29
AF  535 DATA 5,5,5,5,7,7,5,0,39
NR  540 DATA 5,5,5,2,5,5,5,0,32
HJ  545 DATA 5,5,5,2,2,2,2,0,23
MA  550 DATA 7,1,2,2,2,4,7,0,25
RQ  555 DATA 7,4,4,4,4,4,7,0,34
EZ  560 DATA 4,4,2,2,2,1,1,0,16
LR  565 DATA 7,1,1,1,1,1,7,0,19
BJ  570 DATA 2,5,0,0,0,0,0,0,7
EY  575 DATA 0,0,0,0,0,0,15,15,30
HR  580 DATA 0,2,2,7,7,2,2,0,22
FC  585 DATA 0,0,2,1,3,5,3,0,14
TO  590 DATA 4,4,6,5,5,5,6,0,35
LQ  595 DATA 0,0,3,4,4,4,3,0,18
JD  600 DATA 1,1,3,5,5,5,3,0,23
GW  605 DATA 0,0,2,5,7,4,3,0,21
DW  610 DATA 1,2,2,7,2,2,2,2,20
LG  615 DATA 0,0,3,5,5,3,1,6,23
RE  620 DATA 4,4,6,5,5,5,5,0,34
NE  625 DATA 0,2,0,6,2,2,7,0,19
EM  630 DATA 0,2,0,2,2,2,2,4,14
QK  635 DATA 4,4,4,5,6,5,5,0,33
IL  640 DATA 6,2,2,2,2,2,7,0,23
MZ  645 DATA 0,0,6,7,7,5,5,0,30
PC  650 DATA 0,0,6,5,5,5,5,0,26
NC  655 DATA 0,0,2,5,5,5,2,0,19
NO  660 DATA 0,0,6,5,5,6,4,4,30
LF  665 DATA 0,0,3,5,5,3,1,1,18
JX  670 DATA 0,0,5,6,4,4,4,0,23
IH  675 DATA 0,0,3,4,2,1,6,0,16
JA  680 DATA 2,2,7,2,2,2,1,0,18
KO  685 DATA 0,0,5,5,5,5,3,0,23
MD  690 DATA 0,0,5,5,5,2,2,0,19
VM  695 DATA 0,0,5,5,7,7,5,0,29
HZ  700 DATA 0,0,5,5,2,5,5,0,22
OJ  705 DATA 0,0,5,5,5,3,1,6,25
GO  710 DATA 0,0,7,1,2,4,7,0,21
MG  715 DATA 0,2,7,7,5,2,7,0,30
GN  720 DATA 6,6,6,6,6,6,6,6,48
MP  725 DATA 0,14,12,10,1,1,1,0,39
HJ  730 DATA 0,1,3,7,7,3,1,0,22
SW  735 DATA 0,4,6,7,7,6,4,0,34
JF  1000 RESTORE 2000:A=1536
HF  1010 FOR I=1 TO 15:SUMA=0:TRAP 1060
KL  1020 FOR J=1 TO 16:READ X:POKE A,X
HC  1030 SUMA=SUMA+X:A=A+1:NEXT J
HV  1040 READ SPR:IF SPR<>SUMA THEN 1060
RR  1050 NEXT I:END
YL  1060 ? "BLAD W LINII ";I*10+1990
YQ  1070 LIST I*10+1990:? "DANA NR. ";J
FL  1080 END
IB  2000 DATA 104,104,141,24,6,104,141,23
    ,6,104,104,141,247,6,104,104,1463
SO  2010 DATA 141,248,6,104,104,170,173,1
    ,1,141,252,6,169,8,141,201,1866
QL  2020 DATA 6,32,192,6,173,251,6,24,105
    ,124,141,119,6,173,250,6,1614
CN  2030 DATA 141,118,6,173,248,6,141,252
    ,6,169,40,141,201,6,32,192,1872
QO  2040 DATA 6,24,173,247,6,74,8,141,249
    ,6,24,173,251,6,105,129,1622
TD  2050 DATA 168,173,250,6,24,105,80,176
    ,3,76,93,6,200,24,109,249,1742
KC  2060 DATA 6,176,3,76,103,6,200,140,13
    0,6,141,129,6,140,133,6,1401
YL  2070 DATA 141,132,6,160,8,173,1,1,40,
    8,176,4,76,229,6,234,1395
GW  2080 DATA 77,1,1,141,1,1,238,118,6,17
    3,129,6,24,105,40,141,1202
CY  2090 DATA 129,6,141,132,6,176,3,76,16
    0,6,238,130,6,238,133,6,1586
HA  2100 DATA 136,192,0,208,208,104,202,2
    24,0,208,1,96,173,23,6,24,1805
WY  2110 DATA 105,1,141,23,6,144,3,238,24
    ,6,238,247,6,76,22,6,1286
KY  2120 DATA 169,0,141,250,6,141,251,6,1
    60,1,192,0,240,22,24,173,1776
ZY  2130 DATA 250,6,109,252,6,141,250,6,1
    36,176,3,76,202,6,238,251,2108
ZG  2140 DATA 6,76,202,6,96,10,10,10,10,7
    6,131,6,0,0,0,0,639
```

DEMONSTRACJA GRAFIKI

Prezentowany przeze mnie programik, choć bardzo krótki, ma wiele atrakcji i nieograniczone możliwości.

Program ten wykreśla na ekranie fantazyjne wykresy pseudoprzestrzenne. W linie 20 i 50 można wprowadzić własne ciekawsze dane na przykładzie przedstawionej próbki (dodatki I — V). Program mój nie tylko prezentuje możliwości graficzne ósmego trybu, ale również ukazuje przykład praktycznego wykorzystania funkcji matematycznych w Atari.

```
QJ   10 GRAPHICS 24:POKE 710,0:DEG :COLOR 1
RX   20 FR=-85:TR=0:SR=4:PLOT 10,85
QN   30 FOR R=FR TO TR STEP SR
QO   40 FOR I=0 TO 360 STEP 10
SY   50 X=160+SIN(I-R)*(R-65):Y=96+SIN(I*2+R)*(R+96)
UQ   60 DRAWTO X,Y:NEXT I:NEXT R
TQ   70 GOTO 70
```

Opis budowy programu:
Linia 10:
— Przełączenie trybu na graficzny o wysokiej rozdzielczości bez okna tekstowego
— Ustalenie koloru tta i atramentu.
— Ustawienie aigumentów funkcji trygonometrycznych na stopnie (DEG/RAD).

Unia 20:
— iinia z dodatku I — V

Linia 30-60
— pętla zewnętrzna.

Linia 40-60
— pętla wewnętrzna

Linia 50:
— linia z dodatku I — V

Linia 60:
— rysowanie funkcji i zamknięcie pętli

Linia 70:
— "pętla bez końca"

```
XQ 20 FR=20:TR=-95:SR=-5:PLOT 190,113
ZF 50 X=160+SIN(I+R)*(R+139):Y=96+COS(I-R)*R
```

```
RA 20 FR=96:TR=-55:SR=-6:PLOT 160,96
JT 50 X=160+COS(I+R)*(R-100):Y=96+SIN(I*2)*R
```

```
HD 20 FR=-95:TR=20:SR=5:PLOT 160,190
HJ 50 X=160+SIN(3*I)*(R+139):Y=96+COS(I)*R
```

```
PH 20 FR=-85:TR=0:SR=4:PLOT 10,100
TW 50 X=160+SIN(I-R)*(R-65):Y=96+COS(I*2+R)*(R+96)
```

Arnold Adamczyk



ATARI PC2

W bogatej ofercie firmy Atari znajdują się nie tylko komputery XL/XE i ST. Stylizowany znak symbolizujący wulkan Fudżi widnieje dziś także na obudowach komputerów personalnych klasy PC. Atari PC 2 przeznaczony jest dla użytkowników, którzy z różnych względów zamierzają korzystać z oprogramowania IBM PC.

Najważniejszą cech ą Atari PC 2 są duże możliwości dopasowania do wymagań klienta. Do czterech złącz znajdujących się we wnętrzu komputera można dołączyć nowe płytki z układami (karty) realizującymi różne zadania. Wśród dodatkowych kart znajdują się pakiety generujące obraz (karty CGA, EGA, Hercules), układy wejścia/wyjścia, pakiety sieciowe, sterowniki dysku elastycznego, które dobiera się w sposób gwarantujący optymalną realizację zadań na danym stanowisku pracy. Dzięki temu można zadość uczynić specyficznym życzeniom użytkowników w biurze, dla których podstawowym problemem jest przetwarzanie tekstów oraz intensywne prowadzenie naukowego obliczenia.

Częstotliwość procesora wynosi 8 MHz, jednakże niektóre programy przystosowane są do 4,77 MHz. Zdecydowano się więc na rozwiązanie dopuszczające przełączanie pomiędzy tymi dwiema częstotliwościami.

Pamięć operacyjna w podstawowej wersji m 512 KB oraz pamięć obrazu 256 KB. Wystarcza to do pracy z większością programów, jednakże do zadań specjalnych można zażyczyć sobie rozszerzenia do 640 KB.

Do zapisu informacji jak zwykle służy kilka urządzeń. Atari PC2 jest wykonywany w wersji z 2 napędami dysków elastycznych o standardowym formacie 5 1/2 cala i pojemności 360 KB. Mieszczą one prawie 400 stron tekstu, czyli dość pokaźną książkę. Jeśli uznamy tę pojemność za niewystarczającą, to należy zainstalować twardy dysk 30 MB, który pozwala na zapisanie około 17 000 stron maszynopisu.

Klawiatura Atan PC2 odpowiada standardowi AT. Spełnia wymagania ergonomiczności obsługi oraz niezawodności działania. Wydzielone są bloki klawiszy, numerycznych i sterujących kursorem, jednakże zrezygnowano z charakterystycznych dia ST romboidalnych klawiszy funkcyjnych, zastępując je typowym dla innych komputerów rozwiązaniem

Do wyposażenia zestawu należy również mysz. Bez tego prostego urządzenia nie można sobie dziś wyobrazić obsługi programów, szczególnie graficznych.

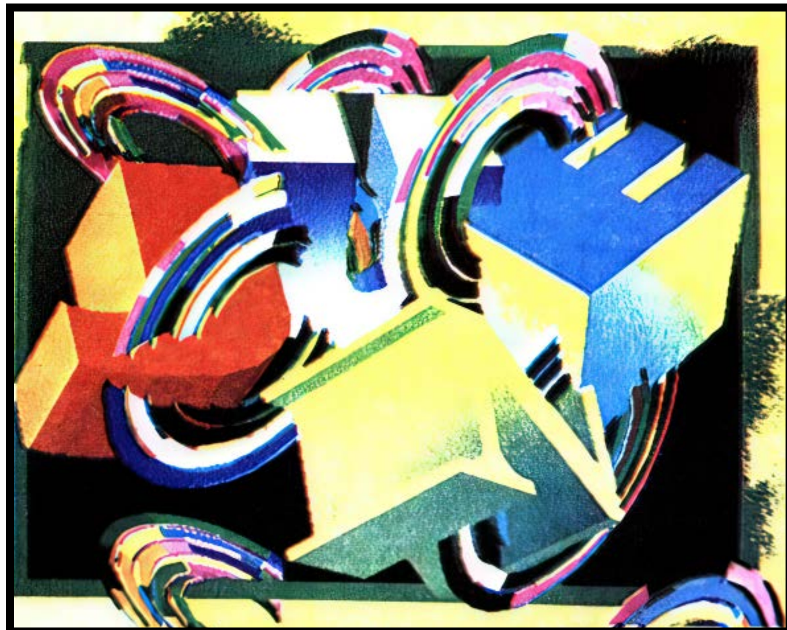
Oferowany wraz z komputerem monitor monochromatyczny ma przekątną 12 cali. W zależności od zastosowanej karty uzyskujemy różne rozdzielczości 720x348 punktów dla Herculesa, 640x350 EGA, 640x200 dla CGA.

Atari PC 2 umożliwia korzystanie z ogromnej biblioteki oprogramowania opracowanego w systemie MS/DOS. Wygodę obsługi polepsza zastosowanie GEM, który pozwala użytkownikowi o niewielkiej wiedzy w pełni wykorzystywać zalety elektronicznego mózgu. Bogate doświadczenia firmy zdobyte przy konstrukcji poprzednich modeli Atari sprawiają, że PC2 jest bardzo udanym komputerem.

(j.j)

DANE TECHNICZNE

— Procesor 8088-2
— Częstotliwość zegara 4.77 lub 8 MHz
— 4 złącza do dołączenia dodatkowych kart 512 KB pamięci RAM z możliwością rozszerzenia do 640 KB
— |2 stacje dysków elastycznych 2x360 KB, twardy dysk 30 MB jako opcja
— Karty graficzne CGA, EGA, EGAM, MDA, Hercules
— Złącza Centronics, RS 232, złącze do dołączenia myszy
— Złącze do dołączenia koprocesora 8087
— W wersji standardowej dostarczane są:
1) Mysz
2) Oprogramowanie MS-DOS 3.21. Basic, GEM



POPRAWKA TURBO–BASIC-a

Do opisywanego już w „Bajtku" interpretera Turbo-Basic XL dołączony jest zazwyczaj kompilator i zbiór zawierający bibliotekę procedur standardowych. Praca z kompilatorem ma niewątpliwie wiele zalet, zdarzają się jednak czasami przykre niespodzianki, w postaci „złego zachowania" niektórych instrukcji.

Przypadek niniejszy dotyczy funkcji **DEC**. Na skutek błędu tkwiącego w kompilatorze, działanie instrukcji A=DEC(A$) daje się opisać w następujący sposób:

A=LEN(A$): A=128+256*A*(A<5)+ + 1024*(A>4).

Dzieje się tak, ponieważ procedura przekształcająca ciąg znaków na odpowiadającą mu 16-bitową liczbę binarną, przekazuje ją pod adresem $D4 na stronie zerowej, pozostawiając w akumulatorze wartość 128, zaś w rejestrze Y długość ciągu (max. 4).

Wykonywana następnie procedura zamiany liczby binarnej na kod BCD, wykorzystuje do obliczeń zawartość rejestrów A i Y (zerując przy okazji obszar $D4—$D9). Pomiędzy tymi dwiema procedurami kompilator „zapomina" wstawić rozkazy:

```
A5 D4   LDA $D4
A4 D5   LDY $D5
```

Błąd ten można usunąć przez ingerencję zarówno w kompilator, jak i w bibliotekę procedur.

Pierwszy z zamieszczonych programów w Turbo-Basicu, tworzy nowy, poprawiony zbiór procedur standardowych, pod nazwą „D:RUNTI- ME2EXE". Drugi program likwiduje inną niedogodność występującą podczas pracy z kompilatorem, umożliwia bowiem łączenie skompilowanego programu z biblioteką procedur w jeden zbiór binarny.

Jacek Żuk

LISTING 1

```
GY 5 GRAPHICS 0
HW 10 ------------------------------
PH 15 TRAP 15:? :? "WLOZ DYSKIETKE
      ZAWIERAJACA ZBIOR":? """D1:RUNTIME.
      EXE"", WCISNIJ RETURN":GET X
FE 20 CLOSE :OPEN #1,4,0,"D:RUNTIME.EXE"
OG 25 BGET #1,$4000,$2A7C:CLOSE
HY 30 ------------------------------
SM 35 POKE $4EBC,$C5:DPOKE $5E92,$FFBE
OY 40 DPOKE $6A7C,$20:DPOKE $6A7E,$A5C0
XR 45 DPOKE $6A80,$A4D4:DPOKE $6A82,$60D5
IA 50 ------------------------------
LJ 55 TRAP 75:? :? "PRZYGOTUJ DYSKIETKE,
      WCISNIJ RETURN":GET X
TT 60 OPEN #1,8,0,"D:RUNTIME2.EXE"
LD 65 BPUT #1,$4000,$2A84:CLOSE :END
IC 70 ------------------------------
OO 75 CLOSE :? :? "BLAD ";ERR:GOTO 55
```

LISTING 2

```
GY 5 GRAPHICS 0
DX 10 DIM A$(190),B$(28),R$(76),FN$(15)
IG 15 ------------------------------
PA 20 FOR I=0 TO 4:READ R$:FOR J=1
   TO 38:A$(38*I+J)=CHR$(DEC(R$(J+J-
   1,J+J))):NEXT J:NEXT I
HY 25 READ R$:FOR I=1 TO
   28:B$(I)=CHR$(DEC(R$(I+I-1,I+I))):NEXT
   I
HY 30 ------------------------------
WM 35 ? :? "WLOZ DYSKIETKE ZAWIERAJACA
   ZBIOR":? """D:RUNTIME2.EXE"", WCISNIJ
   ENTER":GET X:TRAP 170
RV 40 OPEN #1,4,0,"D:RUNTIME2.EXE"
CY 45 BGET #1,$5000,$2A84:CLOSE
IA 50 ------------------------------
NT 55 ? :? "WLOZ DYSKIETKE ZAWIERAJACA
   PROGRAM":? "ZRODLOWY, PODAJ JEGO
   NAZWE:":? :INPUT "D1:",FN$:TRAP 175
KI 60 R$="D1:":R$(4)=FN$
BG 65 OPEN #2,4,0,R$:BGET #2,0,2:IF
   DPEEK(0)<>$C2FB THEN CLOSE :? :? "TO
   NIE JEST ZBIOR ""D:xxxxxxxx.CTB""":GOTO
   55
IC 70 ------------------------------
QW 75 ? :? "PODAJ NAZWE ZBIORU
   WYNIKOWEGO:":? :INPUT
   "D1:",FN$:R$="D1:":R$(4)=FN$
HU 80 TRAP 180:OPEN #1,8,0,R$
GE 85 TRAP 185:BPUT #1,$5000,$01C4:?
   #1;B$(1,4);
PB 90 BPUT #1,$5305,$0BB5:? #1;B$(5,8);
JH 95 BPUT #1,$5EBE,$1BC6:?
   #1;B$(9,12);A$;B$(13,22);
DE 100 BGET #2,$5000,$06:BPUT
   #1,$5000,$06
OX 105 L=DPEEK($5004)-$226F:DPOKE
   $5004,DPEEK($5004)-1:DPOKE
   $5002,$226F
RT 110 BPUT #1,$5002,$04
RP 115 IF L>$6000:BGET
   #2,$5000,$6000:BPUT
   #1,$5000,$6000:BGET
   #2,$5000,L-$6000:BPUT #1,$5000,L-
   $6000
BK 120 ELSE :BGET #2,$5000,L:BPUT
   #1,$5000,L
VP 125 ENDIF :? #1;B$(23);:CLOSE
   :GRAPHICS 0:GOTO 35
OU 130 ------------------------------
WI 135 DATA 85D5A00084D4A204EE01D3B1D4990
   005C8D0F8CE01D3B9000591D4C8D0F8E6D5CAD
   0E560AE24
WP 140 DATA E4AC25E4E8D001C88ED7218CD821A
   9008D0ED478A9FE8D01D3A9CC200006A9E0200
   00684D6A9
KY 145 DATA C085D7A93085D5B1D491D6C8D0F9E
   6D5E6D7A5D7C9CCD0EFA9E485D7A94085D5B1D
   491D6C8D0
PW 150 DATA F9E6D5E6D7D0F3A97E8DFAFFA9218
   DFBFF8DFFFFA9A38DFEFFEE01D3A9408D0ED45
   860A90085
FY 155 DATA 09A50CA40D8D7E208C7F20A97DA02
   0850C840DA9FE8D01D3A2018EF803860 9CA8E4
   4024CEDF9
TY 160 DATA 0030B43B0040C55B0006BD06E202E
   30224068000850 0E002E1029506
PP 165 ------------------------------
CU 170 CLOSE :? :? "BRAK ZBIORU
   ""D1:RUNTIME2.EXE""":GOTO 35
FK 175 CLOSE :? :? "BRAK ZADEKLAROWANEGO
   ZBIORU":GOTO 55
SU 180 CLOSE #1:GOTO 75
CR 185 ? :? "BLAD ";ERR;", SPROBUJ
   JESZCZE RAZ":CLOSE :GOTO 35
```

BLASKI I CIENIE KYAN PASCALA

Wiele się mówi (i słusznie) o zaletach Pascala, jego walorach dydaktycznych oraz łatwości pisania nawet złożonych programów. Praktycznie jedyną dostępną w Polsce realizacją tego języka na Atari jest system Kyan Pascal.

Nie cieszy się on zbytnią popularnością ze względu na konieczność współpracy ze stacją dysków. Pomimo błędów kompilatora, przy pisaniu programów o objętości 300—400 linii jest on najbardziej optymalnym językiem dla tych, którzy chcą tworzyć wygodnie dość szybko wykonywane programy. O ogromnych możliwościach mechanizmu rekurencji czy pascalowskich struktur danych nie trzeba przekonywać nikogo. Chciałbym więc przedstawić krótki opis systemu Kyan Pascal oraz podać kilka uwag, które nasunęły mi się w czasie rocznej pracy z tą realizacją języka.

Typowa praca z kompilatorem obejmuje trzy etapy:
1) utworzenie przy pomocy edytora zbioru zawierającego tekst programu,
2) kompilację programu.
3) uruchomienie programu wynikowego.

Na dyskietce, oprócz DOS, znajdują się następujące zbiory:
AUTORUN.SYS — uruchamiający system Kyan Pascal;
B — pozwala uruchomić system z poziomu DOS;
ED — edytor Kyan Pascala;
PC — kompilator;
LIB — biblioteka procedur standardowych, niezbędna do uruchomienia programu wynikowego;
HELP — podręczna instrukcja obsługi: ponadto dołączony jest pakiet procedur i funkcji — tworzących zbiory z rozszerzeniem .I — oraz programów przykładowych.

Warto zapamiętać, że standardowo kompilator tworzy zbiory wynikowe z rozszerzeniem .0. tak więc gdy np. znajdujący się na dyskietce zbiór PRINT. I zawiera tekst źródłowy programu drukującego teksty, to zbiór PRINT.O zawiera tekst skompilowany tego programu czyli program wynikowy.

PRACA Z PAKIETEM KYAN PASCAL

Po włączeniu komputera (koniecznie z wyłączeniem Basica klawiszem OPTION) możliwe są dwie sytuacje:

— na ekranie pojawia się wizytówka firmy Kyan Software, poniżej zaś znak > (zaproszenie do wydania polecenia). System jest gotowy do pracy.

— zgłasza się DOS, musimy załadować zbiór B — na ekranie również pojawi się znak > oznaczający gotowość systemu do pracy.

Od razu jedna istotna uwaga: pracując z Kyan Pascalem zapomnijmy o klawiszu BREAK. Przypadkowe naciśnięcie go przerywa natychmiast wykonywaną operację „wyrzucając" nas do poziomu systemu (pojawi się >), czasem wręcz zawiesza komputer. Bywa to szczególnie bolesne podczas pisania programu — możemy stracić cały tekst.

Z poziomu systemu Kyan Pascal możemy jedynie uruchamiać gotowe programy z dyskietki:
Dn:ED — edytor;
Dn.PC — kompilatory;
Dn: nazwa.O — program w Kyan Pascalu (zbiór LIB pozostanie wczytany automatycznie);
DOS — wyjść na poziom DOS;
! — uruchomić powtórnie ostatnio wykonywany program.

n oznacza numer stacji dysków. Gdy pominiemy ten parametr, system przyjmie D1:. Posiadacze Atari 130XE mogą korzystać z RAM-dysku jako D8:.

EDYTOR KYAN PASCAL

edytora SpeedScript. Jest to wygodne szczególnie przy dużych programach. Po zgłoszeniu się edytor prosi o podanie nazwy zbioru (Dn:nazwa). Próbując go wczytać — jeżeli takiego zbioru nie ma, edytor poinformuje o tym i po wciśnięciu dowolnego klawisza będzie można pisać nowy tekst.

Klawisz ESC pozwoli nam przejść do MENU edytora. Po wciśnięciu odpowiedniego klawisza możemy:
ESC — wrócić do redagowania programu;
H — wywołać opis klawiszy sterujących;
X — zapisać program do zbioru i wyjść z edytora;
S — zapisać program bez wychodzenia z edytora;
P — zmienić nazwę zbioru (klawisz spacji nie zmienia nazwy);
Q — przerwać pracę i wyjść z edytora tracąc poprawki;
I — wstawić tekst ze wskazanego zbioru począwszy od linii, w której znajduje się kursor;
G — przejść do podanej linii programu;
A —wpisać tekst do ciągu A;
B — wpisać tekst do ciągu B;
C — wymienić w tekście ciągi A i B.

Pozostało jeszcze powiedzieć parę słów o klawiszach sterujących. Oto ich układ:

```
Q W E R T Y
A S D F G
X C V
```

Kursorem sterujemy naciskając klawisz CONTROL i odpowiednią literę.
A — o 1 słowo w lewo;
S — o 1 literę w lewo:
D — o 1 literę w prawo;
F — o 1 słowo w prawo;
E — o 1 linię w górę,
R — o 20 linii w górę;
X — o 1 linię w dół;
C — o 20 linii w dół;
T — na początek tekstu;
V — na koniec tekstu;
G — kasowanie znaku pod kursorem;
Q — kasowanie znaku na lewo od kursora;
Y — kasowanie linii, w któr) znajduje się kursor.

Ponadto działa zwykły edytor ekranowy Atari, Można też kopiować części tekstu:
— ustawiamy kursor na początku tego tekstu i naciskamy CONTROL oraz 0;
— przesuwamy kursor na koniec tekstu i naciskamy CONTROL oraz 0;
— ustawiamy kursor na miejscu, w którym chcemy wpisać oznaczony fragment i naciskamy CONTROL i P.

Po napisaniu i nagraniu programu na dyskietkę wracamy na poziom systemu i wczytujemy kompilator

KOMPILACJA

Edytor ten jest niewielki, a przy tym dość wygodny w użyciu. Można również pisać programy za pomocą

Po zgłoszeniu się Kompilatora podajemy nazwę zbioru zawierającego tekst programu. O ile nie zrobiliśmy błędów, kompilator zapisze program wynikowy do zbioru o takiej nazwie, lecz z rozszerzeniem .0. Ewentualne błędy będą sygnalizowane na monitorze przez podanie numeru linii programu, treści tej linii i rodzaju błędu wraz z miejscem wystąpienia. Niestety, kompilator poprawnie sygnalizuje tylko pierwszy napotkany błąd, potem zachowuje się „nieodpowiedzialnie" podając często wiele bezsensownych komunikatów. Jest to wyraźne niedopracowanie kompilatora, dlatego radzę po otrzymaniu pierwszego komunikatu o błędzie przerywać kompilację klawiszem BREAK. Spowoduje to wyjście na poziom systemu. Możemy teraz wczytać edytor, poprawić błąd itd.

Podając kompilatorowi nazwę zbioru, możemy też podać mu kilka poleceń. Wygląda to na przykład tak: PC D:PROGRAM-L-EP-OD:PROGRAM1.PAS a oznacza:
L — wypisz na monitorze protokół z kompilacji;
LP — wypisz protokół z kompilacji na monitorze i na drukarce;
E — wypisz błędy na monitorze (zawsze włączone);
EP — wypisz błędy na monitorze i na drukarce;
ODn: (nazwa) — utwórz zbiór wynikowy o podanej nazwie, gdy nie podamy nazwy, zbiór nie zostanie utworzony.

URUCHOMIENIE PROGRAMU WYNIKOWEGO

Z poziomu systemu Kyan Pascal podajemy po prostu nazwę zbioru z programem wynikowym. System wczyta wówczas zbiór LIB oraz nasz program. Z poziomu DOS musimy wczytać kolejno LIB i nasz program (możemy połączyć oba zbiory opcją COPY/APPEND z DOS). Jeżeli program nasz przerwie działanie (błąd w programie lub naciśnięty BREAK), otrzymamy komunikat o przyczynie przerwania pracy oraz adres powrotu do programu, zawartość rejestru statusu procesora oraz zawartość akumulatora.

DODATKOWE INSTRUKCJE WYSTĘPUJĄCE W KYAN PASCAL

#A — umieszczone w pierwszej kolumnie nowego wiersza sygnalizuje początek tekstu w asemblerze 6502. Jego koniec oznaczamy znakiem # umieszczonym w pierwszej kolumnie. Ponadto cały ten fragment musi zawierać się między BEGIN a END, np:

```
PROCEDURĘ PAUZA
BEGIN
#A
        LDY #100
TUTAJ DEY
        BNE TUTAJ
#
END;
```

Oto kilka uwag:
1) od pierwszej kolumny zaczynają się tylko etykiety;
2) nie należy zaczynać etykiet literą L — używa ich kompilator;
3) rozpoznawane są następujące dyrektywy:
ORG — umieszczenie kodu wynikowego począwszy od podanego adresu;
EQU — przypisanie etykiecie liczby;
DB — definicja bajtu;
DW — definicja dwubajtowego słowa:
> — mniej znaczący bajt słowa (>$4F80 wynosi $ 0080);
< — bardziej znaczący bajt (<$4F56 wynosi $ 4F00).
4) zdefiniowane są etykiety:

```
SP      EQU 4    wskaźnik początku stosu
LOCAL   EQU 2    wskaźnik końca stosu T EQU 16
```

5) program wynikowy wykorzystuje rejestr X. Używając go użytkownik musi zachować jego stan.

#I (nazwa) — w czasie kompilacji w tym miejscu zostanie wstawiony tekst ze zbioru o podanej nazwie. Uwaga: #I musi być wpisane na początku linii.

SEEK (f,n) — ustawia zmienną buforową, na n-ty element pliku f, czyli pozwala przeczytać (zapisać element pliku f o numerze n. Element pierwszy ma numer 0.

ASSIGN (i,komórka) — przydziela zmiennej i podaną komórkę pamięci. Tak więc możemy zapisać.

```
PROCEDURĘ poke (k.l:integer);
VAR komórka: ^char;
BEGIN
        ASSIGN (komórka.k);
        komórka ^: = 1
END;
```

Wykonanie w programie instrukcji poke (756,204) da taki sam efekt, jak w Basicu.

CHAIN (<nazwa-programu>) — uruchomienie innego programu z zachowaniem obszaru pamięci zmiennych. Jeżeli w tym programie zmienne zostaną zadeklarowane w takiej samej kolejności co w programie, w którym nastąpiło wywołanie, to zostanie im przydzielony ten sam fragment pamięci, więc i odpowiednie wartości.

Dobrze jest wiedzieć, że zmienne typu REAL zajmują 8 bajtów, typu INTEGER — 2 bajty, CHAR — 1 bajt, BOOLEAN — 1 bajt, wskaźniki — 2 bajty a struktury odpowiednią wielokrotność tych wielkos-

FUNKCJE NIESTANDARDOWE

Na dyskietce Kyan Pascal znajduje się zestaw zbiorów z rozszerzeniem I zawierających dodatkowe procedury i funkcje, które można dołączać do własnych

programów:

LENGTH (s) — odpowiednik LEN (S$) w Basicu;

CONCAT (s1,s2,s3) — odpowiednik s3$ = s1$ + s2$;

INDEX (s1, s2) — wartością funkcji jest pozycja, od której łańcuch s1 zaczyna się w s2;

SUBSTRING (s1, k, l) — wartością jest łańcuch długości l wycięty począwszy od k-tego znaku s1;

GRAPHICS (a) — odpowiednik instrukcji GRAPHICS w Basicu. Uwaga przed wywołaniem procedury należy ustawić

RAMTOP (106)na $40 — POKE (106,64); przed GRAPHICS (0) ustawić ponownie na $CO — POKE (106,192).

SETCOLOR (r, h, l), POSITION (x,y), LOCATE (x,y-,d),

SOUND (k,t,b,g) — jak w Basicu;

PLOT (x,x,c) DRAWTO (x,y,c) - jak w Basicu z tym, że c określa kolor;

RANDOM — wartością funkcji jest liczba pseudolosowa z przedziału <0,1).

CECHY SZCZEGÓLNE KYAN PASCALA

1. Program wynikowy nie może przekroczyć objętości 29 kB. W praktyce,gdy wykorzystujemy rekurencję — nawet 20 kB.
2. Obszary pamięci;
   Program wynikowy — od $2000
   Stos programu — poniżej programu, rośnie do $93FF
   Stos arytmetyczny — maleje od $93FF do $2000
   UWAGA: Może nałożyć się na stos programu powodując zablokowanie komputera! Zbiór LIB —$9400-$BFFF.
3. Rozpoznanych jest 8 pierwszych znaków nazwy
4. Zbiory mogą zawierać do 256 elementów.
5. Plikizewnętrznesąpodłączaneautomatycznie,np.
   PROGRAM Test (DANE, WYNIKI);
   ...tekst programu...
   będzie komunikował się z plikami dyskowymi zapisanymi w zbiorach D1 :DANE i D1 :WYNIKI.
6. Instrukcja wiążąca WITH nie działa dla tablic rekordów.
7. Kompilator tylko częściowo sprawdza zgodność typów, np.:
   VAR karta: (trefl, karo, kier, pik);
   BEGIN ...
   IF karta=sigma THEN ...
   zostanie skompilowane pomimo błędu.
8. Brak jest kontroli ilości parametrów w wywołaniu procedur czy funkcji, np.: PROCEDURE Plus (a,b,c integer); może być wywołana jako Plus (12,3), jak też Plus (2,3,4,5). Oczywiście efekt może być bardzo zaskakujący.
9. Nie można indeksować tablicy znakami. Co prawda kompilator przetłumaczy taką deklarację, lecz próba np. wpisania czegokolwiek do elementu tablicy spowoduje błąd.
10. Ostrożnie i tylko w krótkich programach należy używać komend NEW i DISPOSE — ich wykonanie może zablokować komputer (stąd należy unikać używania typu wskaźnikowego).

PODSUMOWANIE

Zalety Kyan Pascala:

- wygodny edytor;
- tłumaczenie tekstu programu na kod procesora;
- możliwość tworzenia procedur w asemblerze;
- dołączony zestaw procedur i funkcji.

Wady:

- błędy w kompilatorze;
- zbyt mała pamięć dla programu;
- konieczność wczytywania edytora w razie wykrycia błędów przez kompilator.

Pomimo wszystko, praca z Kyan Pascalem może dostarczyć sporo satysfakcji. Polecam go wszystkim użytkownikom Atari szukającym wygodnego narzędzia do tworzenia bardziej złożonych programów. Należy jednak być przygotowanym nie tylko na własne błędy w programie, ale i na niespodzianki że strony samego Kyan Pascala.

Witold Rudolf

JESZCZE O ZAMIANIE LICZB

Po zamieszczeniu w „Bajtku" 4/88 programu „Zamiana liczb" otrzymaliśmy wiele listów z podobnymi programami. Na ich opublikowanie trzeba by poświęcić cały numer naszego pisma. Wybraliśmy więc tylko jeden z nich jako przykład.

(red.)

Program przekształcenia liczb w trzech systemach (dwójkowym, dziesiętnym i szesnastkowym) napisałem już dość dawno. Zmusiła mnie do tego konieczność przygotowania różnych rysunków do druku w trybie graficznym drukarki Atari 1029. Niezbędna jest do tego zamiana bajtów (odczytanych w formie binarnej z papieru milimetrowego) na liczby dziesiętne, które można zapisać w liniach DATA programu drukującego. Gdy zacząłem rozważać ten drobny problem, postanowiłem potraktować go kompleksowo — napisać program, który dokonywałby zamiany między wszystkimi trzema systemami. Program został napisany w języku Turbo-Basic, nie tylko ze względu na jego szybkość, lecz przede wszystkim z powodu strukturalnych konstrukcji zaczerpniętych z Pascala.

Algorytm użyty do zamiany liczb dziesiętnych na dwójkowe jest odmienny od algorytmu Doroty Ciesielskiej ("Bajtek" 4/88). Na początku programu w jednowymiarowej tablicy B(16) jest umieszczone szesnaście potęg liczby 2 (od 20 do 215). W zależności od wielkości zadanej liczby dziesiętnej jest ona potem dzielona przez poszczególne elementy tablicy B przy użyciu funkcji MOD, która zwraca resztę z dzielenia. Najlepiej pokaże to przykład:

Mamy liczbę 77. Ponieważ jest mniejsza od 256, to odpowiadająca jej liczba dwójkowa będzie miała osiem cyfr (jedynek lub zer). Dzielenie zaczniemy więc od liczby 128 (t.j. 27, czyli B(7)). przy czym część ułamkową wyniku mniejszego od 1 pomijamy.

77 : 128 = 0
77 : 64-1, reszta 13
13 : 32 = 0
13 : 16 = 0
13 : 8 = 1, reszta 5
5 : 4 = 1, reszta 1
1 : 2 = 0
1 : 1 = 1

Rezultatem jest binarna liczba 01001101. Jeśli zadana liczba dziesiętna jest większa od 255, dzielenie rozpoczyna się od liczby 32768 (t.j. 215, czyli B(15), a wynikiem będzie 16-cyfrowa liczba dwójkowa.

Odwrotna zamiana (z systemu dwójkowego na dziesiętny) jest znacznie prostsza. Kolejne cyfry podanej liczby binarnej (zera i jedynki) mnożymy przez elementy tablicy B (od B(0) do B(15)), przy czym wykonujemy to od prawej do lewej, a wyniki dodajemy. Liczba 01001101 zostanie zamieniona na postać dziesiętną następująco: $1\times2^0+0\times2^1+1\times2^2+1\times2^3+0\times*2^4+0\times2^5+1\times2^6+0\times2^7=77$

Do zamiany liczb w systemie szesnastkowym użyłem funkcji DEC i HEX$, które są implementowane w Turbo-Basicu.

Liczby podawane są komputerowi tak, jak się je pisze, a więc w systemie szesnastkowym np. A01F, przy czym program sam dba, aby nie były przekroczone dozwolone parametry (wartość do 65535, liczby dwójkowe do 16 cyfr, a szesnastkowe do 4 cyfr). Nie są natomiast sprawdzane błędnie wprowadzone cyfry, np. 2 w liczbie dwójkowej, lub H w liczbie szesnastkowej. Zamiany są wykonywane przez procedury DEBI (liczba dziesiętna na dwójkową), BIDE (dwójkowa na dziesiętną). DEHE (dzisiętna na szesnast- kową), HEDE (szesnastkowa na dziesiętną), BIHE (dwójkowa na szesnastkowa) i HEBI (szestnastkowa na dwójkową). Dzięki zastosowaniu Turbo-Basica program jest bardzo czytelny i nie wymaga dodatkowego opisu.

Petr Trnka

```
0 REM  ***********************
1 REM  *    ZAMIANA  d-$-%    *
2 REM  *   Petr Trnka, 1988   *
3 REM  * Copyright (c) Bajtek *
4 REM  ***********************
5 ---------------------------
10 GRAPHICS  0:SETCOLDR 1,0,12:SETCOLOR 2,12,2:SETCOLOR  4,12,2:POKE
752,1
20 DIM H$(5),B(16),B$(17),A(16)
30 FOR N=0 TO 15:B(N)=2^N:NEXT N
40 # START:CLS :A=0
50 # PAS:FOR N=0 TO 39:POSITION N,A:PUT 18:NEXT N:ON A=2 GO# TYTUL
60 A=2:GO# PAS
70 # TYTUL:POSIT1ON  9,1:?  "Z  A  M  I  A  N  A   d-t-%"
80 ? :POSITION 7,4:? "(Wartosc  maksym.:  65535)":? :? :? :?
90 ? " 1  dziesietna -> dwojkowa":?
100 ? " 2  dwojkowa -> dziesietna":?
110 ? " 3  dziesietna -> szesnastkowa": ?
120 ? " 4  szesnastkowa  ->  dziesietna": ?
130 ? " 5 dwojkowa -> szesnastkowa":?
140 ? " 6 szesnastkowa -> dwojkowa":?
150 # WYBOR:GET KEY
160 # SKOK:ON KEY-48 EXEC DEBI,BIDE,DEHE,HEDE,BIHE,HEBI
170 ON KEY-48 GO# BIN,DCM,HEX,DCM,HEX,BIN:GO# WYBOR
200 ---------------------------
210 PROC DEBI
220   # E1:CLS :B$="":? "Zamiana dziesietnej na dwojkowa:":EXEC
PASEK
230   INPUT "Liczba  dec.: ",X:X1=X
240   IF X>65535 THEN EXEC NADMIAR:GO# E1
250   EXEC DB
260 ENDPROC
300 ---------------------------
310 PROC BIDE
320   # E2:CLS :B$="":? "Zamiana  dwojkowej  na dziesietna:":EXEC
PASEK
330   INPUT "Liczba bin.: ",B$
340   IF LEN(B$)>16 THEN EXEC NADMIAR:GO# E2
350   EXEC BD
360 ENDPROC
400 ---------------------------
410 PROC DEHE
420   # E3:CLS :B$="":? "Zamiana dziesietnej na szesnastkowa:":EXEC
PASEK
430   INPUT "Liczba dec: ",X
440   IF X>65535 THEN EXEC NADMIAR:GO# E1
450   H$=HEX$(X)
460 ENDPROC
500 ---------------------------
510 PROC HEDE
520   # E4:CLS :B$="":? "Zamiana szesnastkowej na dziesietna:":EXEC
PASEK
530   INPUT "Liczba hex.: ",H$
540   IF  LEN(H$)>4  THEN EXEC NADMIAR:GO# E4
550   X=DEC(H$)
560 ENDPROC
600 ---------------------------
610 PROC BIHE
620   # E5:CLS :B$="":? "Zamiana  dwojkowej na szesnastkowa:":EXEC
PASEK
630   INPUT "Liczba  bin.:  ",B$ 640   IF LEN(B$)>16 THEN EXEC
NADMIAR: GO# E5
650   EXEC BD 660   H$=HEX$(X)
670 ENDPROC
700 ---------------------------
710 PROC HEBI
720   # E6:CLS :B$="":? "Zamiana szesnastkowej na dwojkowa:":EXEC
PASEK
730   INPUT  "Liczba hex.: " , H$
740   IF  LEN(H$)>4  THEN EXEC NADMIAR:GO# E6
750   X=DEC(H):X1=X
760   EXEC DB 770 ENDPROC
2000 ---------------------------
2010 PROC NADMIAR
2020   # PUT 253:PUT 253:POSITION 14,10:? "ZBYT WIELKA!":PAUSE 150
2030 ENDPROC
2100 ---------------------------
2110 PROC PASEK
2120   FOR N=1 TO 37:PUT 18:NEXT N:? :?  :?
2130 ENDPROC
2200 ---------------------------
2210 PROC DB
2220   IF X=0 THEN B$="0":GO# KONIEC
2230   IF  X<256:N=7:ELSE :N=15:ENDIF
2240   WHILE N>=0
2250     # DALEJ:W=X/B(N)
2260     IF W=1 THEN A(N)=1:EXIT
2270     IF W<1 THEN A(N)=0:N=N-1:GO# DALEJ
2280     IF W>1 THEN A(N)=1:X=X MOD B(N):N=N-1:GO# DALEJ
2290   WEND
2300   IF W=1 AND N<>0
2310    FOR M=N-1 TO 0 STEP -1:A(M)=0:NEXT M
2320   ENDIF
2330   IF  X1<256:N=7:ELSE :N=15:ENDIF
2340   WHILE  N>=0
2350     M=N+1
2360     FOR R=1 TO M
2370       B$(R,R)=STR$(A(N)):N=N-1
2380     NEXT R
2390   WEND
2400   # KONIEC:ENDPROC
2500 ---------------------------
2510 PROC BD
2520   X=0:N=0:U=LEN(B$)
2530   REPEAT
2540     X=VAL(B$(W,W))*B(N)+X
2550     W=W-1:N=N+1
2560   UNTIL W=0
2570 ENDPROC
3000 ---------------------------
3010 REM  * liczba dwojkowa *
3020 ---------------------------
3030 # BIN:Z=20-LEN(B$)/2
3040 POSITION Z,12:? B$:GO# PYTANIE
3100 ---------------------------
3110 REM  * liczba  dziesietna *
3120 ---------------------------
3130 # DCM:Z=20-LEN(STR$(X))/2
3140 POSITION Z,12:? X:GO# PYTANIE
3200 ---------------------------
3210 REM  * liczba szesnastkowa *
3220 ---------------------------
3230 # HEX:Z=20-LEN(H$)/2
3240 POSITION Z,12:?  H$:GO PYTANIE
3300 ---------------------------
3310  REM * co  dalej ? *
3320 ---------------------------
3330 # PYTANIE:POSITION 6,19:? "1 Nastepna zmiana"
3340 POSITION 6,21:? "2 Powrot do glownego menu"
3350 # KLAWISZ:GET Q
3360 ON Q-48 Go# SKOK,START:GO# KLAWISZ
```



DISC LOGO

Wiele szkół posiada już swe wymarzone Spectrum, które używane jest często do nauki LOGO. Współpraca tego języka z magnetofonem jest bardzo utrudniona ze względu na zapis dużej ilości zbiorów bez nagłówka. Jeżeli chce się do zapisu zbiorów wykorzystać stację dysków, w zasadzie nie jest to możliwe bez przeróbki interpretera.

Poniżej podaję sposób zaadaptowania POLSKIEGO LOGO do współpracy ze stacja dysków TIMEX 3".

POLSKIE LOGO składa się z trzech części. Pierwsza z nich to program ładujący (tzw loader) w języku Basic. Ładuje on i uruchamia drugą część — krótki program w kodzie maszynowym, który to wczytuje trzecią część — program główny napisany również w kodzie maszynowym. Jest to zbiór bez nagłówka o długości 33700 bajtów, a umieszczany jest w pamięci poczynając od adresu 16384.

Najpierw należy podzielić trzecią część LOGO na "ekran"i "resztę" Do tego celu użyć należy programu COPY-COPY, w którym wykonujemy instrukcję **LOAD (6912** i ładujemy trzeci segment LOGO Po wczytaniu ustawiamy parametry nagłówka instrukcją **LET 1 = LOGO$,6912,16384** i nagrywamy go na taśmę zleceniem **SAVE**. Na tym kończy się praca z COPY-COPY — wciskamy RESET.

Teraz umieszczamy przygotowaną wcześniej czystą dyskietkę stroną B do stacji i wpisujemy sekwencję rozkazów:

LOAD "LOGOS": PAUSE 0: PRINT #0; INK 0; PAPER 6; "DISCLOGO WERSJA 3.0 ©JAMSOFT'88": SAVE *"D-LOGO_0" SCREENS

Następnie wczytujemy z taśmy nagrany wcześniej plik — podana sekwencja rozkazów posyła go na dyskietkę. Zerujemy komputer.

Ładujemy do pamięci POLSKIE LOGO. Wykonujemy **.ZAJMIJ 6913**, potem **RED** bez parametru i w edytorze wpisujemy procedury umieszczone na wydruku 1. Po zakończeniu wychodzimy z edytora (**EXT MODE C**) Wykonujemy **RED []** i ponownie z niego wychodzimy. Powracamy teraz do BasiC-a przez **DOŚĆ**. Wykonujemy **SAVE *"D -LOGO_1"CODE 24020,41117** — nagramy w ten sposób na dyskietkę LOGO wraz z potrzebnymi procedurami.

Wpisujemy program pilotujący DISCLOGO (wydruk 2). Uwaga! Program musi być wpisany dokładnie tak, jak na wydruku. Nawet najdrobniejsze zmiany uniemożliwią pracę DISCLOGO.

Program uruchamiamy przez **GOTO 10.** Po komunikacie **0 OK** kasujemy linie **10** i **20**. Otrzymany program nagrywamy na dyskietkę przez **SAVE * "D-LOGO_2"LINE 2**.

Teraz do wpisania pozostaje program ładujący z wydruku 3. Należy go oczywiście nagrać na dyskietkę — **SAVE *"DISCLOGO" LINE 1**.

Mając nagrane wszystkie potrzebne pliki, należy skopiować je ze strony B na stronę A w kolejności podanej na wydruku 4.-choć nie jest to konieczne. Aby zabezpieczyć DISCLOGO przed skasowaniem, należy wykonać **ATTR *"+"p.** Od tej chwili DISCLOGO iest gotowe do pracy — wczytuje się je przez **LOAD "'DISCLOGO"**.

DISCLOGO ma wprowadzone dodatkowe 6 instrukcji, poznawanych po prefixowym #. Są to: **#ładujr. #ładujo, #zapiszr, #zapiszo, #kat, #bas**.

Działanie pierwszych czterech jest analogiczne do działania ich odpowiedników w pierwotnych instrukcjach LOGO. Format wywołania tych procedur jest również taki sam, tzn. **instrukcja "nazwa**.

Komenda **#kat** wyświetla katalog wszystkich plików LOGO. Należy zaznaczyć, iż DISCLOGO tworzy własne, opcjonalne pliki, posiadające rozszerzenie **.RED** dla plików utworzonych przez **#zapiszr** i **.SCR** dla plików **#zapiszo**.

#BAS powoduje powrót do interpretera języka Basic. W DISCLOGO komenda **DOŚĆ** nie zadziała.

Działanie programu jest bardzo proste, oparte na fakcie, że w przypadku LOGO. powrót do interpretera Basic-a powoduje przejście do pierwszej linii programu w języku Basic. Procedury DiSCLOGO powodują umieszczenie w pierwszej linii programu „w tle" numeru linii, w której znajduje się obsługa wywołanej procedury. Umieszczamy za instrukcją REM podprogram maszynowy używany jest do obsługi proceoury 4f #zapiszo. Przechowuje on treść ekranu w buforze 58112, ponieważ powrót do Basic-a powoduje wyczyszczenie ekranu.

DISCLOGO wyposażone zostało jedynie w niezbędne instrukcje, gdyż interpreter POLSKIEGO LOGO zostawia na obszar programów w Basic-u ok 250 bajtów. Wszelkie próby oprogramowania stacji z poziomu kodu maszynowego kończą się niczym, gdyż brak jest szczegółowej dokumentacji, zaś informacje podane w instrukcji obsługi są niepełne i ogólnikowe.

Uwagi i spostrzeżenia;

— po zapisie pliku, DISCLOGO kataloguje dysk, by potwierdzić prawidłowość zapisu,

— jeśli przy zapisie pliku podana zostanie nazwa dłuższa niż 8 znaków, uwzględnione zostanie jedynie pierwsze 8,

— DISCLOGO ze względu na ograniczoną ilość pamięci nie posiada odpowiedniej obsługi błędów Jeśli taki wystąpi, powrót do programu następuje przez **GOTO 2**,

— nie można rozbudowywać programu pilotującego, gdyz dodatkowa linia uszkodzi interpreter LOGO,

— DISCLOGO nie może być startowane automatycznie, gdyz wtedy nie działa. Tak naprawdę, to sam nie wiem, dlaczego,

— procedura zapisu ekranu potrzebuje bufora wielkości 7 KB. Dla LOGO pozostaje więc 8 KB pamięci. Jeśli jest to niewystarczające przy tworzeniu **"D-LOGO-2"** należy pominąć komendę **.ZAJMIJ 6913** i wpisać procedury za wyjątkiem **#zapiszo**.

— zalecane jest stosowanie **.ODSMIEĆ**, jeżeli zarezerwowana jest pamięć na bufor ekranu,

Jacek Marczewski

```
Wydruk 1 : Procedury w LOGO

OTO #ZAPISZR :n
C 23863 23870
N :n 23863 23870
.UMIEŚĆ 23762 53 DOŚĆ
JUż

OTO #ZAPISZO :n
C 23947 23954
N :n 23947 23954
.WYWOłAJ 23975
.UMIEŚĆ 23762 55 DOŚĆ
JUż

OTO #LADUJR :n
C 23835 23842
N :n 23835 23842
.UMIEŚĆ 23762 52 DOŚĆ
JUż

OTO N :s :n :a
JEŚLI :n = :a + 1 [stop]
JEŚLI PUSTE :S [stop]
.UMIEŚĆ :n ASCII PIERW :s
N BP :s :n + 1 :a
JUż

OTO C :a :k
JEŚLI :a = :k + 1 [stop]
.UMIEŚĆ :a 32 C :a + 1 :k
JUż

OTO #łADUJO :n
C 23908 23915
N :n 23908 23915
.UMIEŚĆ 23762 54 DOŚĆ
JUż

OTO #KAT
.UMIEŚĆ 23762 51 DOŚĆ
JUż

OTO #BAS
.UMIEŚĆ 23762 57 DOŚĆ
```

```
Wydruk 2 : Pilot w BASIC'u

1 GO TO VAL " " 0B
2 RANDOMIZE USR VAL "24835" E5
3 CAT *"+.red": PAUSE VAL "0" : CLS
: CAT *"+.scr": PAUSE VAL "0" : CLS :
GO TO VAL "2" 70
4 LOAD *"12345678.red"CODE : GO TO
VAL "2" A6
5 SAVE *"12345678.red"CODE VAL
"50402",VAL "2000": GO TO VAL "3" 71
6 LOAD *" 12345678.scr "SCREEN$: GO
TO VAL "2" B0
7 RANDOMIZE USR VAL "23937": SAVE *"
12345678 .scr"SCREEN$ n: GO TO VAL
"2" 07
8 REM   WPISZ TU 24 SPACJE F5
10 FOR A=23975 to 23999: READ B :
POKE A,B: NEXT A 8A
20 DATA 33,0,64,17,0,227,1,0,27,237,
176,201,33,0,227,17,0,64,1,0,27,237,
176,201 7C
```

```
Wydruk 3 : Loader do DISCLOGO

1 BORDER 7: INK 7: CLS :
LOAD *"d-logo_0"SCREEN$ : LOAD
*"d-logo_1"CODE:CLS: PRINT INK
0;PAPER 6;"DISCLOGO JACKE MARCZEWSKI
@1988": LOAD *"d-logo_2" DA
```

```
Wydruk 4 : Poprawny katalog

:DISCLOGO
Level 0   Drive A

Name      Typ     Size  Alloc S P
-----------------------------------
DISCLOGO           129    1K
D-LOGO_0          6917    7K
D-LOGO_1         41122   41K
D-LOGO_2           262    1K

Max 140K   Cur 50K   Rem 90K
```

ERRATA

W siódmym numerze „Bajtka", w artykule „Spróbujmy inaczej" została błędnie wydrukowana tabelka. Za pomyłkę przepraszamy Czytelników i Autora, a wersję poprawną drukujemy obok. red

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	195	110	92	34	31	198	176	205	91	2	205	8	92
1	188	42	201	176	31	91	92	89	205	205	148	205	111
2	91	75	62	92	31	34	115	91	89	1	91	148	96
3	0	92	42	62	31	85	40	205	91	22	255	91	62
4	62	119	215	247	56	92	8	89	205	42	229	225	16
5	30	33	0	219	243	201	35	91	94	176	86	126	205
6	50	228	205	254	205	205	114	24	91	92	43	245	148
7	57	91	160	31	172	160	205	227	24	229	94	35	91
8	92	34	91	48	91	91	172	205	251	62	235	34	62
9	62	85	235	38	33	42	91	172	62	1	62	176	21
	827	909	303	201	924	199	48	398	203	832	418	293	904
0	205	245	91	42	190	9	92	25	97	14	24	58	0
1	140	62	68	75	201	205	205	0	58	0	0	32	0
2	91	23	77	92	197	181	110	238	32	0	238	34	22
3	241	215	205	35	213	3	25	34	34	56	34	59	91
4	254	241	43	35	229	225	68	32	59	91	32	110	0
5	32	215	45	35	17	209	77	108	110	0	65	58	13
6	56	215	205	94	50	193	201	105	58	13	100	249	255
7	1	201	227	35	0	201	0	99	249	0	114	192	0
8	215	205	45	86	33	42	0	122	192	0	101	48	29
9	201	140	201	175	0	73	1	98	48	1	115	14	32
	436	762	207	704	130	341	779	861	937	175	823	854	442

FOLLOW ME

Tytuł ten, w wolnym tłumaczeniu, może znaczyć "rób to, co ja". Pomysł zaczerpnięty jest z francuskiego czasopisma „Amstrad Magazine"; program podobny figurował pod nazwą „Garfunkel". Tu jednak został znacznie skomplikowany, co zapewne czyni go atrakcyjniejszym, lecz trudniejszym.

Na ekranie widnieje kolorowe koło, podzielone na osiem jednakowych części, ponumerowanych od 0 do 7. Komputer wybiera losowo którąś z części i zakrywa ja przez chwiię. Zadaniem grającego jest powtórzenie tego ruchu, przez wciśnięcie jednej z cyfr od 0 do 7. Po prawidłowej odpowiedzi komputer losuje następną część koła i zakrywa po kolei: najpierw poprzednią, potem wylosowaną. W ten sposób trudność gry wciąż rośnie.

Na początku nie będziesz mógł zapamiętać nawet pięciu zniknięć, po krótkim treningu z łatwością powtórzysz osiem. Jednak wynik piętnastu zniknięć należy uznać za bardzo dobry, nawet po długim treningu.

Stanisław Dałek

```
  10 REM **********************
  20 REM *** S & D  Company ***
  30 REM ***                ***
  40 REM ***  FOLLOW   ME   ***
  50 REM **********************
  60 REM
  75 LET k=0: REM 31
  80 CLS : REM 51
  85 RANDOMIZE : REM 54
  87 IF PEEK 65000<>17 THEN GO
SUB 1000: REM -Mcode- : REM 55
  90 GO SUB 330: REM  -
Rysunek- : REM E0
 100 GO SUB 560: REM  - Dane -:
REM 7B
 104 BRIGHT 1: REM 91
 105 PRINT AT 5,11;"FOLLOW ME":
REM 41
 106 PRINT AT 8,0;" Na
ekranie widnieje kolorowe
kolo. Jego fregmenty
beda       znikac w wyznaczonej
przez      Komputer kolejnosci.
": REM 95
 108 PRINT " Twoim
zadaniem jest powtorzenie
sekwencji zniknie c, poprzez
wystukanie odpowiednich
cyfr    na klawiaturze
": REM 6D
 109 BRIGHT 0: REM 94
 110 PRINT AT 21,0;" Nacisnij
klawisz aby zaczac ": PAUSE 0:
REM 72
 112 GO SUB 330: REM 5B
 114 DIM t(100): REM 35
 115 LET n$=" Zaczynamy ": GO
SUB 950: PAUSE 100: GO SUB 860:
REM 6F
 130 LET n=1: REM 6D
 140 LET n$=" Najpierw  ja. ":
GO SUB 950: REM B8
 145 PAUSE 100: REM 95
 150 LET t(n)=INT (RND*8): REM
36
 160 GO SUB 630: REM - Beep  -:
REM FF
 170 GO SUB 860: REM B1
 190 FOR i=1 TO n: REM DA
 195 LET n$=" A teraz ty ": GO
SUB 960: REM 07
 200 PAUSE 0: REM 05
 210 LET a$=INKEY$: REM 35
 225 LET n$=" A teraz ty ": GO
SUB 950: REM 25
 260 IF a$<"0" OR a$>"7" THEN
LET n$=" Jak mam to rozumiec ?!
": GO SUB 950: PAUSE 50: GO SUB
860: GO TO 195: REM FD
 270 LET j=VAL a$: GO SUB 700
 275 IF VAL a$<>t(i) THEN GO TO
900: REM 38
 280 NEXT i: REM 7C
 310 LET n=n+1: REM BD
 320 GO TO 140: REM 6B
 330 REM --- rysunek  ---:  REM
A9
 335 IF PEEK 65009=235 THEN
RANDOMIZE USR 65000: RETURN :
REM AE
 340 RESTORE 510: REM EC
 350 FOR i=-80 TO 80: REM 8F
 360 LET X=SQR (6400-i*i): REM
4E
 370 PLOT 123-x,i+86: REM DF
 380 DRAW 2*x+.5,0: REM 9F
 390 NEXT i: REM EA
 400 INVERSE 1: REM BB
 410 FOR i=85 TO 90: PLOT 40,i:
DRAW 180,0: NEXT i: REM C4
 420 FOR i=120 TO 127: PLOT
i,0: DRAW 0,175: NEXT i: REM 11
 430 FOR i=40 TO 47: REM 67
 440 PLOT i,8: DRAW 160,160:
REM A2
 450 NEXT i: REM 26
 460 FOR i=200 TO 207: REM 23
 470 PLOT i,8: DRAW -160,160:
REM EE
 480 NEXT i: REM 44
 490 INVERSE 0: REM 13
 500 FOR i=0 TO 7: READ x,y:
PRINT AT y,x;i: NEXT i: REM 6F
 510 DATA 18,3,22,7,22,13,18,17
,12,17,8,13,8,7,12,3: REM EA
 520 LET k=1: REM F2
 530 FOR j=0 TO 7: GO SUB 700:
NEXT j: REM 42
 540 LET k=0: REM 04
 545 LET g=USR 65000: POKE
65009,235: REM 7E
 550 RETURN : REM 2C
 560 REM : REM 22
 570 REM --- Tabele  ---:  REM
D5
 580 RESTORE 610: REM 43
 590 DIM d(8): REM 4C
 600 FOR i=1 TO 8: READ d(i):
NEXT i: REM 36
 610 DATA 0,2,4,5,7,9,11,12:
REM 65
 620 RETURN : REM 72
 630 REM : REM 68
 640 REM  --- Beep   ---: REM
48
 650 FOR i=1 TO n: REM A8
 660 LET j=t(i): REM 68
 670 GO SUB 700: REM FF
 680 NEXT i: REM 0D
 690 RETURN : REM B8
 700 REM  :  REM AE
 710 REM --- Miganie   ---:
REM CF
 720 LET a=65092+5*j: REM DB
 730 POKE 65016,INT (a/256):
REM 68
 740 POKE 65015,a-INT
(a/256)*256: REM E0
 750 IF k=1 THEN GO TO 790: REM
6A
 760 POKE 65030,0: REM 81
 770 LET g=USR 65013: REM 6D
 780 BEEP .4,d(j+1): REM 56
 790 POKE 65030,1: REM A2
 800 LET g=USR 65013: REM 8B
 810 RETURN : REM 31
 820 REM : REM 27
 870 REM --- Zmazuje doi   ---:
REM FF
 880 PRINT #1;AT 0,0;"
": REM 3B
 890 RETURN : REM 81
 900 CLS : REM 88
 910 PRINT AT 4,0;"Bardzo mi
przykro,ale masz        skleroze
!": REM E6
 912 PRINT "Nie martw sie
tak bardzo,bowiem umiesz juz
sekwencje ";STR$ n;" liczb":
REM E0
 913 PAUSE 300: REM 63
 916 GO SUB 330: REM 82
 917 PRINT : PRINT AT 21,0;"
Oto prawidlowa sekwencja":
PAUSE 150: REM 3F
 918 GO SUB 640: REM BF
 920 GO TO 110: REM A4
 950 REM ---Pisze na dole---:
REM CA
 960 PRINT #1;AT 0,0; FLASH
1;n$: REM F8
 980 RETURN : REM DB
1000 REM  --- MCOde   ---: REM
1E
1002 RESTORE 1000: REM 9C
1005 LET s=0: REM DE
1010 FOR i=65000 TO 65131: READ
d: LET s=s+d: REM 3E
1020 POKE i,d: REM F5
1030 NEXT i: REM 6D
1045 IF s<>11747 THEN PRINT
"Zle wpisales linie 1050-1100
!": STOP : REM C2
1050 DATA 17,232,226,33,0,64,1,
0,27,0,237,176,201: REM 61
1060 DATA 221,33,0,0,221,78,0,2
21,102,2,221,110,3,221,94,4,62,
0,254,1: REM 36
1070 DATA 40,2,14,63,221,126,1,
254,0,32,15,22,1,205,50,254,20,
205,62,254: REM E0
1080 DATA 122,254,11,32,244,201
,22,10,205,50,254,21,205,62,254
,122,254,0,32,244: REM 42
1090 DATA 201,62,0,113,35,60,18
6,32,250,125,146,111,201,213,22
,0,25,209,201: REM D6
1100 DATA 57,1,88,16,32,58,0,88
,57,31,59,1,89,112,33,62,0,89,1
44,32: REM 27
1110 DATA 58,0,89,142,31,61,1,8
9,101,32,56,0,88,37,32,60,1,88,
5,33: REM C1
1120 RETURN : REM 68
```

KATALOG

DLA C-16/116 I PLUS/4

Program KATALOG ułatwia odnajdywanie na kasecie programów bez konieczności kontrolowania wskazań licznika obrotów magnetofonu. Wyszukiwanie odbywa się podczas szybkiego przewijania taśmy magnetofonowej i zakończone jest sygnałem akustycznym połączonym z jednoczesnym zatrzymaniem silnika magnetofonu.

Na jednej stronie kasety magnetofonowej typu C-60 można umieścić 10 programów, których czas odczytu nie przekracza 165 sekund. Odpowiada to normalnym programom o pojemności ponad 8 KB lub, gdy korzystamy z programów zapisanych w systemie szybkiego wczytywania (np. TURBO C-16), ich pojemność może przekroczyć 30 KB.

Program KATALOG należy zapisać na samym początku kasety. Oczywiście przed zapisaniem należy w liniach DATA na końcu programu umieścić nazwy programów które będą zapisane na tej kasecie.

Po wczytaniu i uruchomieniu KATALOGU należy wybrać numer interesującego nas programu i włączyć szybkie przewijanie. Magnetofon zatrzyma się w miejscu zarezerwowanym na wybrany przez nas program.

KATALOG jest przerobioną wersją programu zamieszczonego w YOUR COMMODORE 06/86.

Tadeusz Chęciński

```
3C  100 REM ***   KATALOG   ***
38  105 :
D1  110 REM ***  T.CHECINSKI  ***
F8  115 :
78  120 :
0F  125 CS$=CHR$(147):RV$=CHR$(18):RO$=CHR$(146)
E7  130 S$=CS$+RV$+"   *** MASTER TAPE PROGRAM
        CATALOG ***  "+RO$
35  135 X=10:P$=" PROGRAM ":Q=205:K=64784
B5  140 PRINT S$:POKE Q,5
4F  145 FOR I=1 TO X:READ A$:PRINT P$+A$:NEXT
F6  150 POKE Q,20: INPUT" PODAJ NUMER PROGRAMU:
        ";A
02  155 IF A<1 OR A>X GOTO 150:ELSE POKE 0,7
D2  160 PRINT S$;SPC(83);"WLACZ SZYBKIE
        PRZEWIJANIE DO PRZODU,"
03  165 PRINT:PRINT"  WCISNIJ DOWOLNY KLAWISZ."
86  170 GETKEY C$:IF PEEK(K)=255 GOTO 160
5D  175 PRINT S$;SPC(90);CHR$(130);" PROSZE
        CZEKAC !";CHR$(132);SPC(56)
39  180 POKE 0,15:TI$="000000":B=(A-
        L)*(13.1-A/3)
66  185 C=TI/60
B1  190 PRINT," JESZCZE";INT(B-C);CHR$(157);"
        SEK.":POKE Q,5
99  195 IF C<B GOTO 185:ELSE POKE 0,7
A0  200 PRINT S$;SPC(89)"ZATRZYMAJ MAGNETOFON"
42  205 VOL 8:SOUND 1,850,20:FOR T=1 TO 90:NEXT:
        SOUND 2,800,20
A6  210 IF PEEK(K)=255 GOTO 215:ELSE GOTO 210
CC  215 PRINT;SPC(170);"WYKONAJ LOAD/SAVE"
26  220 RESTORE:FOR I=1 TO A:READ A$:NEXT
E1  225 PRINT SPC(45);P$+A$:POKE 0,15
4F  230 :
25  235 REM TU WPISZ NAZWY PROGRAMOW
10  240 :
30  245 DATA "1 ...............
40  250 DATA "2 ..............."
12  255 DATA "3 ..............."
22  260 DATA "4 ..............."
F3  265 DATA "5 ..............."
0C  270 DATA "6 ..............."
0D  275 DATA "7 ..............."
ED  280 DATA "8 ..............."
BF  285 DATA "9 ..............."
A4  290 DATA "10..............."
```

ALLOCATOR

Zapisywanie programów na uszkodzonej dyskietce jest co najmniej ryzykowne — zawsze może się zdarzyć tak, że fragment programu zostanie wpisany do sektora niepewnego czy wręcz uszkodzonego. Takiej „przyjemnostki” można jednak uniknąć stosując program podany poniżej.

Poszczególne ścieżki i sektory dyskietki możemy bez obawy rezerwować do własnych potrzeb za pomocą instrukcji BLOCK-ALLOCATE. Jej format jest następujący:

100 OPEN 15,8,15
120 PRINT # 15, "B-A"; 0; ŚCIEŻKA;SEKTOR
140 CLOSE 15

Jeżeli zamierzasz Czytelniku poeksperymentować trochę z tą instrukcją, to radzę wziąć dyskietkę, która nie zawiera ważnych dla Ciebie programów.

Taki sposób rezerwowania wybranych sektorów do własnych potrzeb jest najlepszym z możliwych na dyskietce dobrej. Problem polega tu na tym, że procedura ta jest niedopracowana i nie zawsze chce działać tak jakbyśmy sobie tego życzyli.

Poniższy program pozwala na zarezerwowanie dowolnego sektora na wybranej ścieżce z wyjątkiem ścieżek systemowych (tzn. przeznaczonych dla katalogu dyskietki). Program działa ze wszystkimi stacjami Commodore. Test nośnika możemy przeprowadzić za pomocą programu DISK TESTER podanego w tym numerze.

Na zakończenie dwie uwagi. Po pierwsze pamiętajmy, że wykonanie instrukcji VALIDATE lub COLLECT automatycznie spowoduje uwolnienie zajętego w ten sposób sektora. Po drugie zarezerwowanie uszkodzonego sektora znacznie obniża lecz nie likwiduje całkowicie ryzyka utraty programu — jeżeli np. uszkodzenie to było czysto mechaniczne, osłabiony nośnik może stopniowo odklejać się dalej powodując błędy w innych, wcześniej dobrych sektorach. Dlatego też proponuję, aby na takich dyskietkach NIGDY nie zapisywać programów dla nas ważnych. Pamiętajmy także o tym, że proces ten powoduje zmniejszanie się liczby sektorów wolnych na dyskietce; innymi słowy, jeżeli zarezerwowaliśmy cztery uszkodzone sektory na pustej dyskietce to przy odczytywaniu katalogu otrzymamy „660 BLOCKS FREE” zamiast „664 BLOCKS FREE”.

Klaudiusz Dybowski

```
1E  100 REM ***    ALLOCATOR    ***
38  105 :
AF  110 REM ***   K. DYBOWSKI   ***
F8  115 :
78  120 :
68  125 CLS$=CHR$(147):UP$=CHR$(145): DN$=CHR$(17)
27  130 PRINTCLS$DN$DN$
1E  135 INPUT" STACJA 1541, 1570 CZY 1571
    (4/0/7):";DT:PRINTDN$DN$
CB  140 IF DT=7 THEN OPEN15,8,15,"UO>M1":CLOSE
    15:GOTO 155
47  145 IF DT=4 OR DT=0 THEN OPEN
    15,8,15,"UO>MO":CLOSE 15:GOTO 155
08  150 RUN
89  155 INPUT" NPISZ NUMER SCIEZKI: ";TR
68  160 IF TR<1 OR TR=18 OR TR=53 OR TR>70 THEN
    PRINTUP$UP$:GOTO 155
00  165 IF TR>35 THEN W=35
73  170 PRINTDN$:INPUT" WPISZ NUMER SEKTORA: ";SC
42  175 IF SC<0 OR S>20 THEN PRINTUP$UP$:GOTO 170
35  180 IF TR<=17+W AND SC>20 GOTO 170
DF  185 IF TR=>19+W AND TR<=24+X AND SC>18 GOTO 170
F6  190 IF TR=>25+W AND TR<=30+W AND SC>17 GOTO 170
F2  195 IF TR=>31+W AND TR<35+W AND SC>16 GOTO 170
13  200 IF TR<=35 THEN TK=4*TR
05  205 IF TR>35 THEN TK=3*(TR-36)-1
A5  210 IF SC=0 OR SC<=7 THEN PT=TK+1:PR=0:GOTO 225
D0  215 IF SC>7 AND SC<=15 THEN PRINT
    PT=TK+2:PR=8:GOTO 225
45  220 PT=TK+3:PR=16
B7  225 BIT=2^(SC-PR)
58  230 OPEN 15,8,15
58  235 PRINT#15,"IO"
5B  240 OPEN 5,8,5,"#"
89  245 PRINT#15,"U1";5;0;18+W;0
49  250 PRINT#15,"B-P";5;PT
1E  255 GET#5,B$:IF B$="" THEN B$=B$+CHR$(0)
7A  260 BYTE=ASC(B$)
1D  265 G=BYTE AND BIT
C0  270 IF G=0 THEN PRINTDN$DN" TEN SEKTOR JEST JUZ
    ZAJETY !":GOTO 365
78  275 BYTE=ASC(B$)AND(255-BIT)
0F  280 PRINT#15,"B-P";5;PT
74  285 PRINT#5,CHR$(BYTE);
41  290 IF TR>35 THEN PRINT#15,"U2";5;0;53;0:GOTO
    325
64  295 PRINT#15,"B-P";5;TK
A6  300 GET#5,Q$:IF Q$="" THEN Q$=Q$+CHR$(0)
BA  305 Q=ASC(Q$)-1
75  310 PRINT#15,"B-P";5;TK
75  315 PRINT#5,CHR$(Q);
78  320 PRINT#15,"U2";5;0;18+W;0:GOTO 360
DF  325 PRINT#15,"U1";5;0;18;0
D7  330 PRINT#15,"B-P";5;221+(TR-36)
08  335 GET#5,T$:IF T$=" THEN T$=T$+CHR*(0)
A1  340 T=ASC(T$)-1
D5  345 PRINT#15,"B-P";5;221+(TR-36)
4E  350 PRINT#5,CHR$(T);
8A  355 PRINT#15,"U2";5;0;18;0
C4  360 PRINTDN$DN$" SEKTOR";SC;"NA
    SCIEZCE";TR;"ZAREZERWOWANY."
95  365 CLOSE 5:PRINT#15,"IO':CLOSE 15
3D  370 END
```

DOBRZE I ŹLE O STACJI 1571

1571 może, jak wiadomo, pracować aż w trzech trybach pracy: jednostronnym (emulacja doskonale wszystkim znanej stacji 1541), dwustronnym i w systemie MFM co pozwala odczytywać programy pracujące pod kontrolą CP/M. Posiadacze 1571 i C-64 czy któregoś z komputerów rodziny C-16 dość często zwracali się do nas listownie z prośbą o wyjaśnienie czy jest możliwe wykorzystanie dwustronnego zapisu gdy stacja ta pracuje z innym niż C-128 komputerem.

Poniżej podany sposób ma swoie zalety i wady. Do zalet należy lepsze wykorzystanie dyskietki, obrót samego dysku tylko w jedną stronę co na pewno przedłuża żywotność nośnika i oczywiście brak konieczności przekładania dyskietki na drugą stronę gdy chcemy skorzystać z programu tam zapisanego. Główną wadą tego sposobu jest niemożność odczytania programów zapisanych na drugiej stronie dyskietki jeżeli nie korzystamy przy tym ze stacji 1571.

Po uruchomieniu komputera i stacji tryb pracy dwustronnej włącza się następująco: OPEN 15, NUMER, 15, "U O > MI": CLOSE 15 gdzie NUMER oznacza numer urządzenia (od 8 do 11).

Teraz kolej na dyskietkę. Oczywiście zapis dwustronny jest możliwy wyłącznie na odpowiednio sformatowanej dyskietce. W tym celu wykonujemy:

OPEN 15, NUMER,15, "NONAZWADYSKU.ID":CLOSE 15

jeżeli mamy C-64 lub HEADER "NAZWA DYSKU", IID

gdy korzystamy z C-16/116 lub PLUS 4.

Po ok. minucie proces formatowania zostanie zakończony. Po wczytaniu katalogu dyskietki (directory) do pamięci i jego wyświetleniu powinniśmy otrzymać w rezultacie 1328 bloków wolnych zamiast 664. Dyskietka jest gotowa do zapisu...

Stosując ten sposób należy pamiętać o kilku zasadach:

1. Przed każdym odczytaniem lub zapisem programu (a zwłaszcza w tym ostatnim wypadku!) konieczne jest przełączenie stacji na dwustronny tryb za pomocą powyżej podanego polecenia. Także i inne operacje takie jak kasowanie programu, czy "czyszczenie" dyskietki (VALIDATE, COLLECT) Muszą być wykonywane w dwustronnym trybie pracy.

2. Korzystanie z dwustronnego trybu pracy stacji z komputerem innym niż C-128 NIE daje możliwości korzystania z opcji szybszego wczytywania programów. Gdy nasza dyskietka jest już zapisana do końca warto ją zabezpieczyć przed przypadkowym skasowaniem poprzez założenie naklejki zabezpieczającej na wycięcie w boku dyskietki. Innym sposobem może być zmiana jednego z bjtów na ścieżce 18 w sektorze 0: trick ten polega na zmianie numeru systemu operacyjnego na dyskietce co stacja traktuje jako nośnik zapisany na innej stacji (np. 4040). W efekcie ROM 1571 uniemożliwi zapis czy jakiekolwiek inne operacje do chwili sformatowania dyskietki na nowo lub do momentu posłużenia się odpowiednim programem odbezpieczającym.

Z mojej praktyki wynika przede wszystkim, że stosowanie tego sposobu przynosi dość wymierne korzyści w zakresie wykorzystania przestrzeni użytkowej dyskietki. Jeżeli przy zapisie jednostronnym zostawało nam zwykle ok. 20-30 bloków wolnych z obu stron to na dyskietce zapisanej dwustronnie otrzymujemy do dyspozycji ich sumę; zwykle wystarcza to do zapisania jeszcze jednego średnio długiego programu.

Klaudiusz Dybowski

O @SAVE SŁÓW KILKA

Jeszcze w podręczniku stacji 1571 można znaleźć na stronie 26 informację na temat SAVE WITH REPLACE czyli zastępowania starego programu na dyskietce jego nową wersją pod tym samym tytułem. Tymczasem już dawno dawno temu, kiedy zabawa w mikrokomputery dopiero się zaczynała instrukcja ta znalazła się pod obstrzałem krytyki użytkowników.

Gdzieś koło 1984 roku pewien Kanadyjczyk przyjął prace zlecone, które postanowił wykonać za pomocą swojego Commodore 64 i stacji dysków 1541. Aby ułatwić sobie pracę stosował on właśnie te instrukcję. W efekcie dość drogo go to kosztowało. Oto bowiem podczas demonstracji jednego z trzech gotowych programów okazało się, ze dwa istnieją a trzeci gdzieś zniknął, pomimo iż jego tytuł jest zapisany w katalogu dyskietki. Koniec końców okazało się, ze dyskietka jest w porządku ale jej mapa (BAM) jest nieźle pokręcona.

Ponieważ sygnałów takich zaczęło nadchodzić coraz więcej, jedno z czasopism zwróciło się do firmy Commodore z prośbą o wyjaśnienie na które się niestety nie doczekało. Wobec tego Czytelnicy przejęli ster w swoje ręce i zabrali się do studiowania systemu operacyjnego stacji 1441. Na efekty nie trzeba było długo czekać

Po dokładnej analizie okazało się, że nie wszystkie procedury zawarte w ROM są dopracowane do końca. Nieporządek wprowadzały tu przede wszystkim procedury obsługujące bufory pamięci RAM; gdy były one zapełnione zdarzało się, że do BAM dyskietki wpisywane były zupełnie nieaktualne informacje dotyczące ścieżek i sektorów wykorzystywanych przez zapisywany za pomocą SAVE@ program. W efekcie tytuł się zgadzał tyle tylko, że program był zapisany w zupełnie innym obszarze dyskietki niżby to wynikało z BAM

Pomimo tego okazało się, że nawet zerowanie buforów stacji (OPEN 15,8,15, .10": CLOSE 15) nie jest w stanie zawsze zapobiec błędom w zapisie. Wykryto przy tym, że błędy takie powstają także wtedy, gdy przy SAVE@ nie podaje się numeru stacji (0 lub 1). Tyle wykazało śledztwo czytelników, którego wyniki publikowało swego czasu wiele czasopism; wyniki te przesłano firmie Commodore.

Większa awantura wybuchła dopiero wtedy, gdy okazało się, że nowe dziecko Commodore — stacja 1571 — ma wspominane wyżej procedury nie poprawione (z wyjątkiem jednej odnoszącej się do numeru stacji). Przekonałem się o tym. osobiście, gdy ufnie stosując SAVE "@: NAZWAPROGRAMU" straciłem bezpowrotnie cztery godziny pracy. Stąd też chciałby zabezpieczyć Czytelników BAJTKA przed takimi niespodziankami.

Jeżeli zdecydowałeś się już na korzystanie z SAVEA@ to pamiętaj aby:

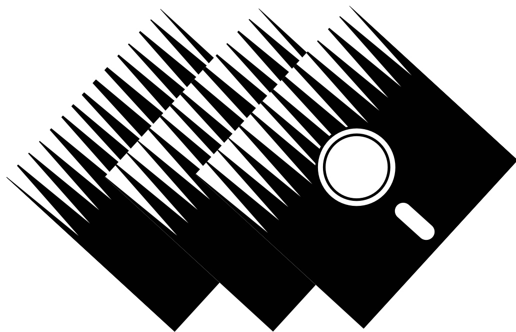
1. Przed każdym zapisem tego typu przeprowadzić inicjalizację stacji w sposób podany powyżej lub (lepiej) wyłączając ją na chwilę z sieci.

2. ZAWSZE dodawaj numer stacji do tego polecenia (czyli stosuj SAVE "@0: NAZWA PROGRAMU" a nie SAVE" NAZWA PROGRAMU").

3. Jeżeli zależy Ci bardzo na Twoich zbiorach czy programach zapisuj je zawsze pod tytułem tymczasowym; następnie skasuj stary program i zmień tytuł tymczasowy na stały W ten sposób unikniesz kłopotów jakie jest w stanie ściągnąć na Ciebie stosowanie SAVE@.

4. Stosowanie się do powyższych zasad nie oznacza wcale, że SAVE@ nie pokaże Ci kiedyś swojej ciemnej strony.

Klaudiusz Dybowski



LOCKER

LOCKER jest programem umożliwiającym zabezpieczenie dyskietki przed zapisem w sposób programowy.

W efekcie jego działania w katalogu dyskietki na ścieżce 18 w sektorze 0 zmieniana jest wartość drugiego (licząc od zera) bajtu określającego system operacyjny stacji dysków na której dyskietka ta była zapisywana. Normalnie wartość tego bajtu wynosi 65; wartość ta jest zamieniana przez program na wartość 233

Polecałbym ostrożne i rozważne korzystanie z tego programu, gdyż zmiana ta może mieć charakter trwały, jeżeli użytkownik nie dysponuje odpowiednim programem odblokowującym, gdyż jedynym możliwym odblokowaniem w takim wypadku jest sformatowanie dyskietki i zapisanie jej od nowa. Dlatego też radziłbym, aby blokować w ten sposób jedynie te dyskietki, na których z całą pewnością nie będziemy chcieli nic więcej zapisać. Dotyczy to także takich operacji jak VALIDATE czy RENAME. UWAGA. Zabezpieczenie to NIE NISZCZY dyskietki ani nie chroni jej przed sformatowaniem. Ponadto niektóre programy kopiujące są w stanie obejść to zabezpieczenie.

Klaudiusz Dybowski

```
E0  100 REM   ***    LOCKER    ***
38  105 :
AF  110 REM   *** K.DYBOWSKI   ***
F8  115 :
78  120 :
5B  125 Y$=CHR$(17)
46  130 PRINT CHR$(147)
29  135 PRINT" TEN PROGRAM ZABEZPIECZA
    DYSKIETKE"Y$
6F  140 PRINT" PRZED PRZYPADKOWYM
    ZAPISEM. ODBLO-"Y$
30  145 PRINT" KOWANIE JEST MOZLIWE
    POPRZEZ SFOR-"Y$
6D  150 PRINT" MATOWANIE DYSKIETKI LUB
    UZYCIE"Y$
7A  155 PRINT" SPECJALNEGO PROGRAMU."Y$
3B  160 PRINTY$Y$
72  165 PRINT" WLOZ DYSKIETKE I WCISNIJ
    RETURN."Y$
02  170 PRINT
    175 GET Z$:IF Z$<>CHR$(13) GOTO 175
AF  180 OPEN 15,8,15,"I0"
49  185 OPEN 5,8,5,"#"
77  190 PRINT#15,"U1";5;0;L8;0
BE  195 PRINT#15,"B-P";5;2
46  200 PRINT#5,CHR$(255);
B0  205 PRINT#15,"U2";5;0;18;0
EA  210 CLOSE 5
BB  215 PRINT#15,"I0":CLOSE15
64  220 PRINTY$" DYSKIETKA
    ZABEZPIECZONA."
E4  225 END
```

PORADY SPOD LADY

W moim magnetofonie uszkodziła się głowica. Od czego mogła się ona uszkodzić i czy można ją wymienić na polski odpowiednik? Zaznaczam, że korzystam tylko z kaset chromowych o wysokiej jakości.

Nazwisko i adres do wiad. redakcji.

Przyczyn uszkodzenia głowicy może być bardzo wiele. Najczęściej spotykane uszkodzenie (?) to po prostu, wytarcie jej czoła przez przesuwającą się taśmę. Czasami może tu pomóc odpowiednia regulacja, czasami jednak trzeba element ten wymienić na nowy. Zastąpienie oryginalnej głowicy polską jest możliwe z tym, że fachowiec powinien dobrać głowicę o zbliżonej indukcyjności. Takiej wymiany można w zasadzie dokonać w każdym punkcie usługowym.

W omawianym przypadku podejrzewam, że to właśnie jest przyczyną uszkodzenia. Stosowanie kaset chromowych do magnetofonów Commodore mija się i z sensem i z celem. Po pierwsze taśma chromowa jest znacznie "twardsza" niż zwykła czy typu FERRO; jej stosowanie skraca więc znacznie żywotność głowicy. Po drugie prawidłowy zapis na takiej kasecie jest możliwy przy określonym prądzie podkładu, którego DATASSETTE NIE jest w stanie wytworzyć. Stąd tez moja rada do wszystkich — uzywajcie kaset dobrych firm lecz bez przesady...

* * *

Parę miesięcy temu kupiłem na giełdzie dyskietki (SSDD) z których przez pewien okres czasu nie miałem okazji korzystać. Gdy chciałem ostatnio nagrać pewien program okazało się, że dyskietka jest wadliwa na obu stronach. To samo powtórzyło się na 2 innych. Obejrzałem dokładnie obie strony ale nie znalazłem rys ani plam. Co może być przyczyną tych błędów?

Jarek z Warszawy

Wydaje mi się, że jest to przede wszystkim wina samego nośnika, a nie stacji jak to sugerujesz w dalszej części listu. Aby dyskietka była uszkodzona wcale nie trzeba rys czy plam — wystarczy, że warstwa tlenków magnetycznych została położona nierównomiernie czego nie da się stwierdzić gołym okiem. Jeżeli teraz podczas formatowania takiej dyskietki akurat w tym miejscu stacja zapisze np. początek sektora czy ścieżki to wcale nie oznacza to, że informacja ta została zapisana dobrze. Przy weryfikacji zapisu może okazać się, że znaczniki te są zapisane z błędem co uniemożliwia następnie odczytanie takiego sektora czy nawet całej ścieżki.

* * *

Ostatnio poszukiwałem dobrego programu katalogującego programy zapisane na dyskietkach. Polecono mi program o nazwie DISKKART, który okazał się niestety uszkodzony. Usterka w jego działaniu pojawiła się gdy wpisałem ponad 1200 programów. Na ekra nie ukazała się nagle nazwa programu wyświetlona zupełnie z boku. Wszelkie próby jej usunięcia kończyły się skasowaniem wcześniej wczytywanych pozycji.

Wadę tą ma wiele innych programów, nie tylko DISKKART. Na to samo natknąłem się pracując z programami SPEEDSORT V5.0 oraz MULTIPROG. Wszystkie te programy zapisują dane bezpośrednio w komórkach pamięci. Aby zwiększyć jej pojemność wykorzystywany jest w nich obszar od adresu 40 960 do 49 151 czyli pamięć "leżąca" pod ROM BASIC. Niestety procedury sortowania i zapisu danych w pamięci są szybkie ale za to niedokładne. Efekty zna Pan z własnych doświadczeń. W wypadku DISKKARTU jedyną metodą jaką udało mi się zastosować z powodzeniem był natychmiastowy zapis całego zbioru i wyłączenie komputera. Po takiej operacji zbiór tytułów programów wczytany do pamięci był nietknięty. Z dobrych programów katalogujących polecałbym DCMR (DISK CATALOG MANAGER/REPORTER); słyszałem również wiele dobrego o programie DISCAT choć nie miałem jeszcze okazji zapoznania się z nim bliżej.

* * *

W instrukcji obsługi stacji dysków znalazłem informację o poleceniu ALLOCATE, za pomocą którego można zabezpieczyć przed zapisaniem wybrane sektory lub ścieżki. Czy po wykonaniu ALLOCATE można zapisywać programy na częściowo uszkodzonej dyskietce jeżeli sektory z błędami zostaną zapisane jako "zajęte"?

Marcin z Wrocławia

Odpowiedź na to pytanie jest dosyć trudna. Generalnie rzecz biorąc ALLOCATE dokonuje wpisu do katalogu dyskietki zmieniając zapis "sektor wolny" na "sektor zajęty". Niestety ALLOCATE jest procedurą niedopracowaną (np. stacja 1571 w trybie 1541 zajmuje od razu całą ścieżkę). Wynika więc z tego, że znacznie lepiej byłoby odczytać całą mapę dyskietki (BAM) i zmienić odpowiednio zapis bez uciekania się do pomocy ALLOCATE. Taki program przedstawiam oddzielnie w tym numerze BAJTKA. Warto także pamiętać, że wykonanie VALIDATE lub polecenie BLOCK-FREE usuwa ten zapis, jeżeli sektor nie jest zajęty naprawdę..

Klaudiusz Dybowski

DISK–TESTER

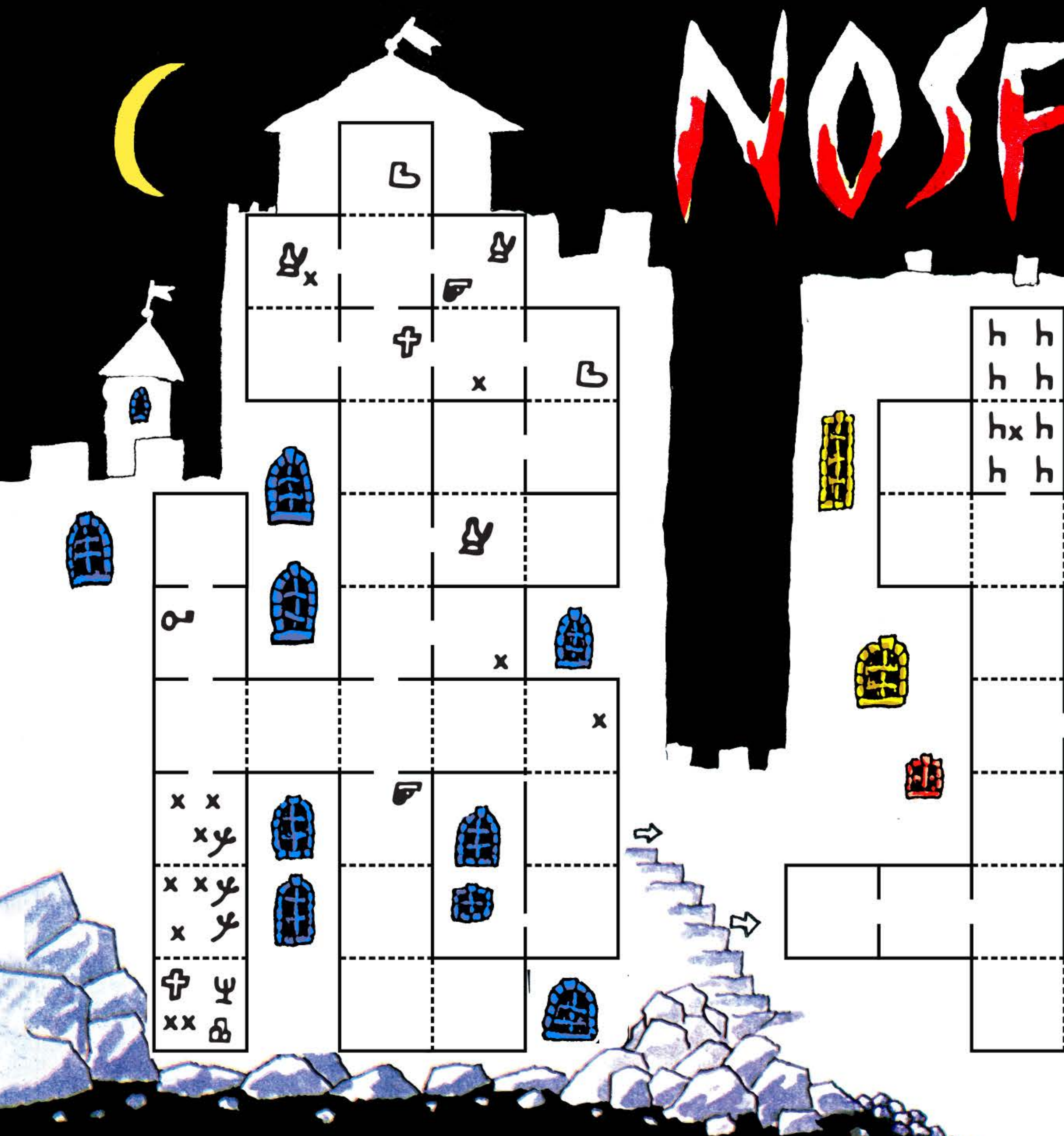
Opisany tu program nie wymaga w zasadzie żadnych specjalnych rekomendacji — umożliwia on przetestowanie dyskietki i wykrycie jej ewentualnych uszkodzeń lub mankamentów nośnika. Program ten można wykorzystywać na wszystkich stacjach Commodore.

Sam tekst dyskietki jest nieco obostrzony. Normalnie procedura odczytu uszkodzonego sektora jest wykonywana 5 razy natomiast w podanym poniżej programie — tylko raz. Pozwala to na ewentualne wychwycenie wszystkich sektorów „niepewnych", które mogą sprawić kłopoty w niedalekiej przyszłości. Z programu tego można korzystać na dyskietkach już zapisanych gdyż odczytuje on jedynie zawartość poszczególnych sektorów.

Pierwsze pytanie zadawane przez program odnosi się do typu używanej stacji. Jeżeli dysponujesz stacją 1541, 1570 czy 1551 to należy wybrać opcję 35 ścieżek, gdyż modele te pracują w oparciu o zapis jednostronny. Jeżeli natomiast korzystasz ze stacji 1571 to wybieraj 70 ścieżek jeśli dyskietka jest zapisana dwustronnie oraz 35 ścieżek jeśli jest ona zapisana jednostronnie.

Klaudiusz Dybowski

```
F4  100 REM   ***  DISK TESTER  ***
38  105 :
AF  110 REM   ***  K. DYBOWSKI  ***
F8  115 :
78  120 :
62  125 D$=CHR$(145):SC=0:SK=0:EK=0
12  130 A$=CHR$(44):C$="  "
B8  135 I$=CHR$(106):J$=CHR$(0):K$=CHR$(1)
81  140 DIM S(70)
3B  145 :
06  150 FOR K=0 TO 35 STEP 35
0D  155 FOR I=1+K TO 17+K:S(I)=20:NEXT
4E  160 FOR I=18+K TO 24+K:S(I)=18:NEXT
03  165 FOR I=25+K TO 30+K:S(I)=17:NEXT
12  170 FOR I=31+K TO 35+K:S(I)=16:NEXT
93  175 NEXT K
FC  180 :
4E  185 PRINT CHR$(147):PRINT:PRINT
3B  190 INPUT" OD KTOREJ SCIEZKI (1)  :
    ";SK
99  195 IF SK=0 THEN SK=1
74  200 IF SK<1 OR SK>70 THEN PRINT
    D$D$:GOTO 190
5E  205 PRINT:PRINT
A1  210 INPUT" DO KTOREJ SCIEZKI (35) :
    ";SC
CD  215 IF SC<0 OR SC>70 THEN PRINT
    D$D$:GOTO 210
5F  220 IF SC=0 THEN SC=35
4A  225 IF SK<1 OR SK>70 THEN PRINT
    D$D$:GOTO 190
4F  230 :
E0  235 PRINT CHR$(147)CHR$(18);
63  240 OPEN 15,8,15,"I0"
D4  245 PRINT#15,"M-W"I$J$K$K$
1A  250 OPEN 5,8,5,"#"
27  255 PRINT" TRACK SECTOR STATUS":PRINT
0E  260 FOR T=SK TO SC
42  265 FOR I=0 TO S(T)
32  270 PRINT#15,"U1";5;0;T;I
B6  275 INPUT#15,E$,F$,G$,H$
2F  280 B$="":IF I<10 THEN B$=" "
58  285 PRINTC$;T;C$B$;I;C$E$A$F$
7F  290 IF E$="00" THEN PRINTD$D$
36  295 NEXT I,T
F5  300 PRINT#15,"M-W"I$J$K$CHR$(5)
60  305 CLOSE 5:CLOSE 15:PRINT:PRINT:END
```



Po długich poszukiwaniach wyprawa archeologiczna, w której i ty bierzesz udział, odkryła wreszcie wejście do piramidy Cheopsa, w której wnętrzu miała ponoć znajdować się mumia faraona. Po odsunięciu płyty Po długich poszukiwaniach wyprawa archeologiczna, w której i ty bierzesz udział, odkryła wreszcie wejście do piramidy Cheopsa, w której wnętrzu miała ponoć znajdować się mumia faraona. Po odsunięciu płyty, z otworu zaczął wydobywać się gęsty, duszący dym. Większość uczestników wyprawy momentalnie zasnęła. Na szczęście Ty zdążyłeś ubrać się w kombinezon filtracyjny chroniący przed wszelkiego rodzaju wyziewami.

Zagłębiasz się więc teraz w czeluść grobowca z zamiarem rozwikłania jego tejemnicy. Zdajesz sobie sprawę z niebezpieczeństwa, lecz jednocześnie wiesz, że powodzenie tej długo przygotowywanej wyprawy zależy teraz wyłącznie od Ciebie. Musisz się bardzo spieszyć, gdyż o odkryciu wejścia, z otworu zaczął wydobywać się gęsty, duszący dym. Większość uczestników wyprawy momentalnie zasnęła. Na szczęście Ty zdążyłeś ubrać się w kombinezon filtracyjny chroniący przed wszelkiego rodzaju wyziewami.

Zagłębiasz się więc teraz w czeluść grobowca z zamiarem rozwikłania jego tejemnicy. Zdajesz sobie sprawę z niebezpieczeństwa, lecz jednocześnie wiesz, że powodzenie tej długo przygotowywanej wyprawy zależy teraz wyłącznie od Ciebie. Musisz się bardzo spieszyć, gdyż o odkryciu wejścia dowiedzieli się już rabusie wyspecjalizowani w ograbianiu piramid. Stary Cheops wiedział, że ludzi pociąga bardziej złoto niż mumia, umieścił więc w piramidzie kilkanaście cennych wyrobów ze szczerego złota. Porozrzucane są one po zakamarkach piramidy i dotarcie do nich jest bardzo trudne. Wszędzie czyhają pułapki w postaci kolców, zapadni, śmiercionośnych ścian a także jadowitych skorpionów.

Rozwikłanie tajemnicy grobowca nie jest jednak możliwe bez zebrania wszystkich skarbów, gdyż tylko wtedy można otworzyć sarkofag i bezpiecznie opuścić piramidę.

Pyramania to jeszcze jedna gra z serii zapoczątkowanej przez JET SET WILLY. Jest ona jednak dużo trudniejsza niż pierwowzór, choć ma tylko 31 komnat. W każde miejsce prowadzi tu w zasadzie jedyna droga, w odkryciu której na pewno pomoże mapa.

Autor: N. MURRAY
Komputer: ZX Spectrum 48/128/+
Amstrad/ Schneider

(mp)

FERATU



NOSFERATU

Butenko pinxit

LEGENDA

 Miecz

 Czosnek

 Krzesło

 Topór

 Buty

 Pistolet

 Krzyż

 Klucz

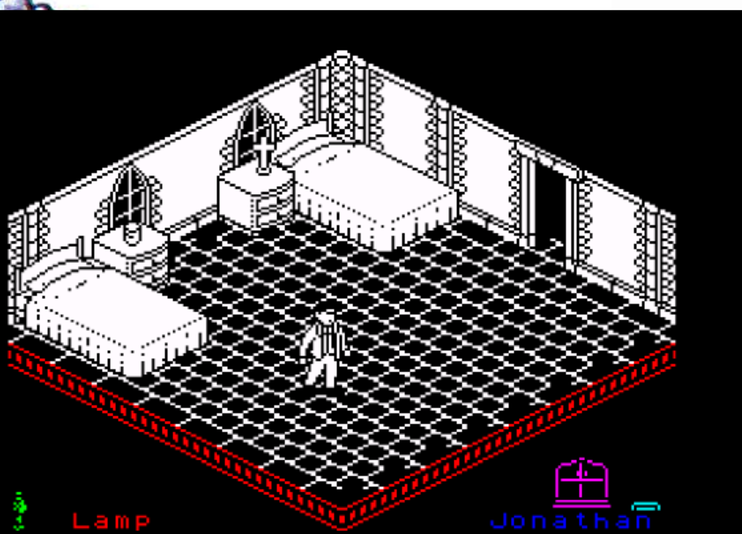
 Zamknięte drzwi

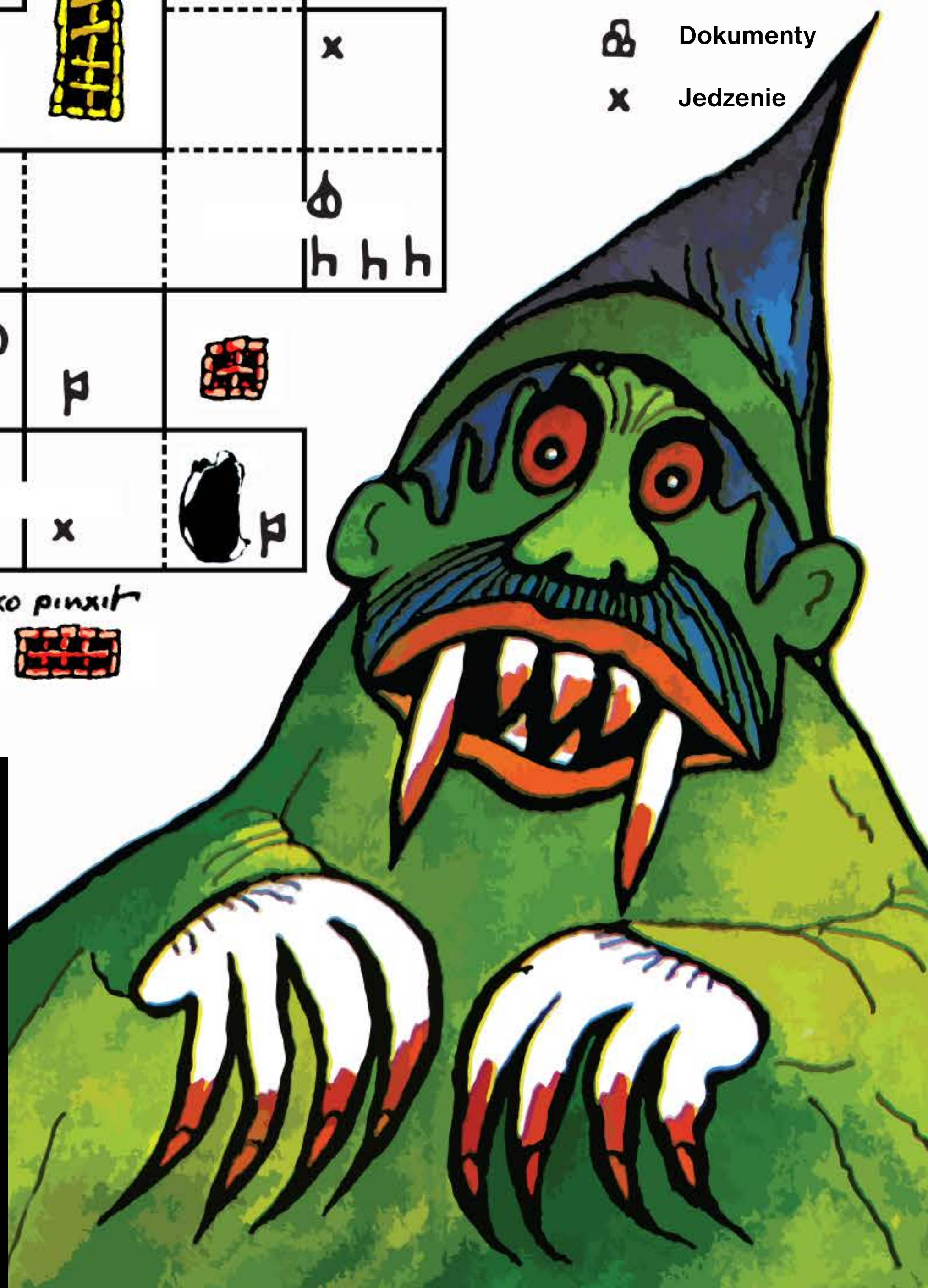
 Lampa

 Świecznik

 Dokumenty

 Jedzenie

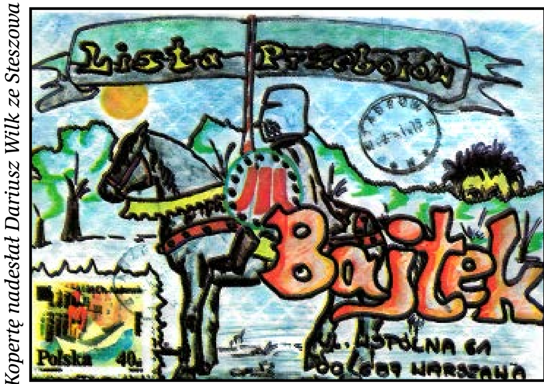




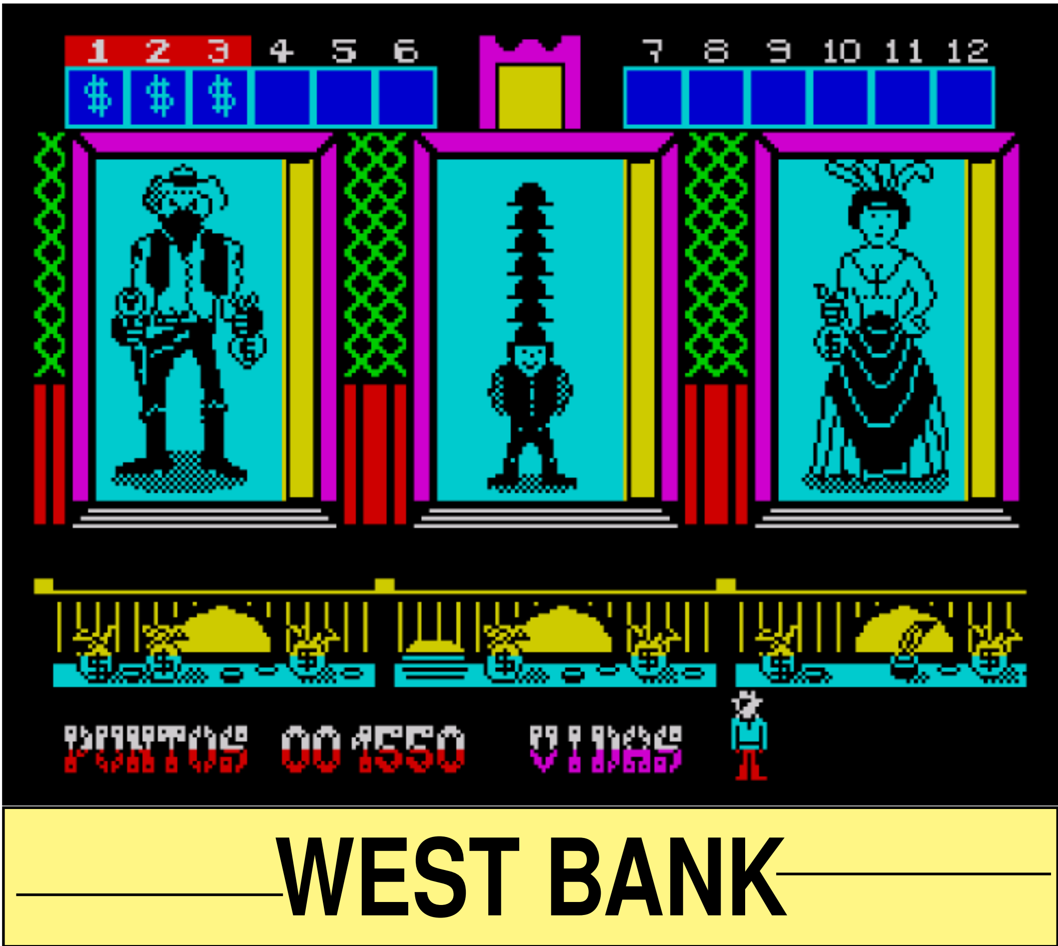
10
BAJTKOWA LISTA PRZEBOJÓW (10/88)

Na pierwszym miejscu tym razem już trzecia część zręcznościowej gry, zatytułowanej "The Way of the Exploding Fist". Wysoko jest też "Sekretny Dziennik Adriana Mole". Zaś "Detective" wszedł od razu na piąte miejsce i szybko nie opuści Listy. Obserwujemy także powrót starego, lecz ciągle dobrego "Stealtha". Tym razem otrzymaliśmy 1625 propozycji na 150 tytułów gier.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	EXPLODING FIST III	↑		x	x	
2	SKYFOX	↑		x	x	x
3	SECRET DIARY	↑		x	x	x
4	URBAN UPSTART	!				x
5	DETECTIVE	!			x	
6	RENEGADE	↓	x	x	x	x
7	ACE	↓			x	x
8	BASKETBALL	!		x	x	x
9	STRIKE FORCE COBRA	↑		x	x	x
10	STEALTH	!	x		x	



Kopertę nadesłał Dariusz Wilk ze Steszowa



WEST BANK

Jadąc na Dzikim Zachodzie zależy od celnego oka i szybkiej ręki. Jest to szczególnie ważne dla przedstawicieli prawa. Jeśli myślisz, że dasz sobie radę, zostań strażnikiem banku.

Bank, w którym pracujesz ma dwanaście-ro drzwi. Naraz czynne są trzy, te, które wybierzesz. Bank czynny jest całą dobę. Ludzie wchodzą i wychodzą załatwiając swoje interesy. Wśród nich kręcą się bandyci. Twoim zadaniem jest unieszkodliwienie rabusiów. Jeżeli będziesz się długo zastanawiał, oni uprzedzą Cię i obrabują bank.

Uważaj dobrze na wchodzące do banku postacie. Są wśród nich damy, które przyszły wpłacić swoje oszczędności, chłopcy ze stertą kapeluszy na głowie (jeśli chcesz trochę zarobić, zestrzel je — czasami jest pod nimi sakiewka z pieniędzmi). Przychodzą także ciemne typki, które dopiero po chwili wyciągają colta. Jeżeli strzelisz do takiego, zanim będzie miał w ręku pistolet, zostaniesz ukarany. Są też tacy, którzy trzymają w ręku sakiewkę, po chwili chowają ją i wyciągają pistolet. Ich też musisz zabić. Jednak najbardziej niebezpieczni są desperaci idący Va Banque! Ci strzelają od razu, zanim zdążysz na nich spojrzeć.

Pod wieczór, gdy wszystkie kasy są pełne, do banku wchodzą trzej kowboje. Chwilę się naradzają i... wszyscy wyciągają pistolety. Ich też musisz zabić, byle nie za wcześnie.

Nadchodzi noc. W mroku nie widać twarzy, tylko zarys sylwetki interesanta. Kręci się więcej bandytów, musisz uważać, gdyż są jeszcze bardziej niebezpieczni

W miarę upływu czasu do miasteczka przyjeżdżają watahy rewolwerowców. Są coraz szybsi, coraz przebieglejsi i jest ich coraz więcej. Nie możesz im ulec, tym bardziej, Ze za wszystko dostajesz sowitą zapłatę. Jest ona tym większa, im bardziej narażone było Twoje życie.

Nie łudź się jednak, że wpiszesz się na listę najlepszych strzelców Dzikiego Zachodu. Najniższy wynik to 241 000, największy to ponad ćwierć miliona. Zdobył go znany nam dobrze kowboj Lucky Lucke, a z nim nikt nie może się równać.

Autor: Alvaro Mateos
Firma: Dinamic Software
Komputer: ZX Spectrum 48/+, Commodore 64/128

(mateo)

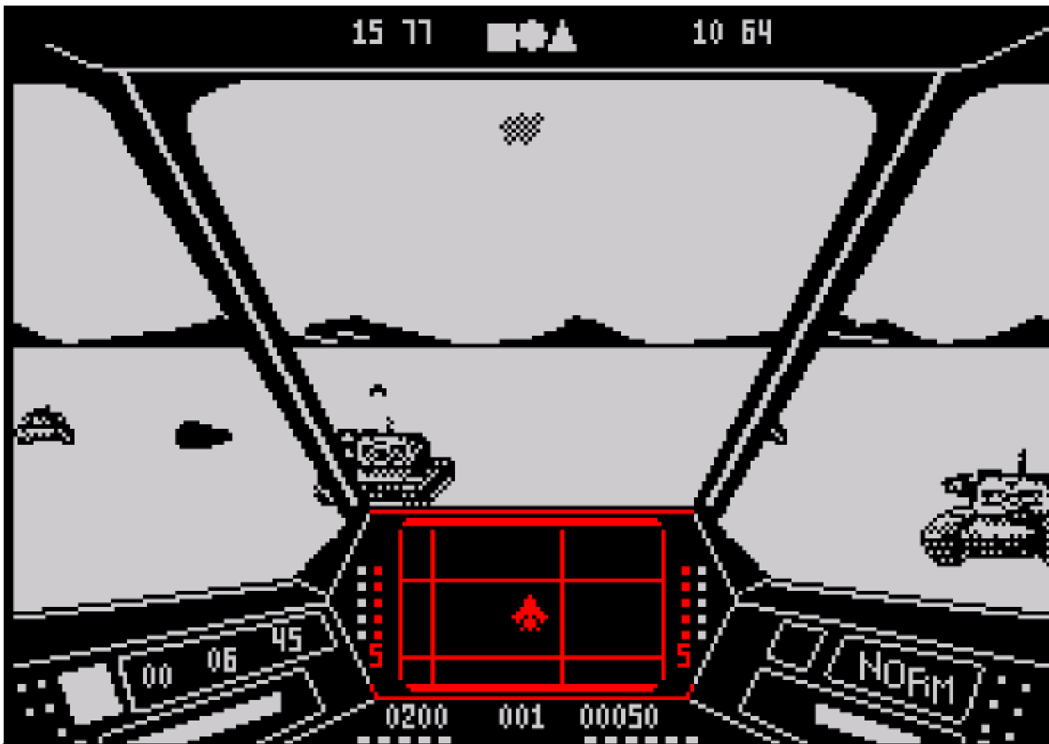
KRÓL I KRÓLOWA GIER



Dominik Spangenberg
uczeń III klasy SP nr 202 w W-wie
posiadany komputer: Atari 65XE
ulubiona gra: Jet Boot Jack
zainteresowania: książki, układanie klocków Lego



Małgorzata Drzewieska
uczennica V klasy SP nr 31 w Gdyni
komputer: Timex
ulubiona gra: Boulder Dash
hobby: sport, w szczególności piłka ręczna



SKY FOX

Obraz przyszłości widziany oczami twórców gier komputerowych jest najczęściej tragiczny. Ziemia została napadnięta, ograbiona, czy wręcz rozbita. Zadaniem gracza jest ratowanie zagrożonej cywilizacji lub podbijanie innych. Co smutniejsze, świat rzeczywiście zmierza w tym kierunku i jak na razie prawie nic nie zapowiada poprawy.

Nie odeszli daleko od standardu twórcy kolejnej „Inteligentnej strzelaniny" — gry SKYFOX (w wolnym) tłumaczeniu — Podniebny Lis). Wcielamy się tu w postać pilota potężnego myśliwca, którego zadaniem jest obrona Ziemi przed najazdem zaborców z głębin kosmosu. Myśliwiec wyposażony jest w oczywiście w najnowocześniejszą broń i sprzęt elektroniczny. Z nowinek technicznych mamy tu radar połączony ze skanerem i noktowizorem, do śledzenia lotu maszyny z góry i przez przednią szybę. Oczywiście cały czas możemy spoglądać do przodu i na boki przez szybę rzeczywistą. Zasłania ją tylko włączenie komputera pokładowego, który oferuje wgląd na mapę terenu i poszczególne jej sektory.

Ziemię atakują dywizje czołgów i eskadry samolotów. Do szybkiego odnalezienia wroga służy autopilot. Nie umie on jednak strzelać, więc wyłącza się po doprowadzeniu samolotu do grup nieprzyjacielskich maszyn.

Myśliwiec wyposażony jest w dwa rodzaje broni — ciężkie działko pokładowe i wyrzutnię rakiet samonaprowadzających się. Obie są równie użyteczne, jeśli chodzi o niszczenie wroga, lecz ich ilość jest ograniczona. Podobnie jest z paliwem. Dozbroić się można w jednej z baz, na powierzchni Ziemi. W dotarciu do bazy pomoże komputer pokładowy.

Gra w SKYFOX to nie tylko latanie i strzelanie bez końca. Początkujący mogą wybrać dwa rodzaje praktyki w odpieraniu ataku czołgów i dwa w zestrzeli- waniu samolotów. Można też ćwiczyć zmasowaną inwazję — małą i dużą — czołgów i samolotów naraz. Właściwa gra to kilka innych wariantów, niewiele się od siebie różniących.

Do wyboru mamy również poziom własnych umiejętności, w postaci kolejnych stopni wojskowych, od stopnia Kadeta do Asa.

Wszelkie opcje wybierane są na początku gry, ruchem joysticka w poziomie lub pionie. Odpowiadają temu przyciski, z góry zdefiniowane. Rozpoczęcie gry następuje po wciśnięciu FIRE (CAPS SHiFT). Przejście do komputera możliwe jest po wciśnięciu C. Tam klawisz H powoduje wyświetlenie listy wszystkich przycisków, zarówno maszyny, jak i komputera. Możliwe jest to w każdej chwili, więc nie będę tu podawał tej listy

Trudno jest rozbić się o Ziemię, chyba, że się na niej usiądzie. W zasadzie myśliwiec uszkodzić mogą jedynie pociski wroga. Wszystko to sprawia, że gra ma w sobie mało z symulatora, polecam ją więc każdemu.

Komputer: ZX Spectrum 48/+, Commodore 64/128

(mp)

DROGA REDAKCJO!

Nazywam się Krzysztof Jasiński, mam 13 lat. Nie posiadam komputera, ale niedługo będę miał Atari 130XE. Chciałbym się wypowiedzieć na temat dosyć długo publikowanych w Bajtku w Klanie Atari pozycji „Tajemnice Atari" oraz „Zostań nieśmiertelnym". W rubryce „Tajemnice Atari" opisujecie (najczęściej) jak przejść bardzo trudne etapy lub skomplikowane miejsca. Czytelnicy Bajtka po przeczytaniu uruchamiają swoje komputery i grają w grę używając Waszych rad. W ten sposób gra staje się nieciekawa, można powiedzieć bez sensu, bez niespodzianek czyhających na gracza w zakamarkach komnat, bez zagadek do rozwiązania. Chciałbym też zwrócić się do czytelników Bajtka w jakim celu piszecie do redakcji, aby opublikowano w rubryce SOS wasze prośby o nadesłanie opisu gry trudnego miejsca, nad którym tak się głowicie? A właśnie to Wy sami powinniście rozgryźć zagadkę następnego etapu, jak wejść po przedmiot potrzebny do otwarcia drzwi, a wreszcie dojść do końca. Korzystając z pomocy psujecie atmosferę gry która jest tak zagadkowa", oraz „nieśmiertelności" tych, którzy lubią prawdziwą grę. Tym, którzy się ze mną zgadzają lub krytykują moje myślenie podaję adres:

Jasiński Krzysztof
ul. Narutowicza 131 m 65
90-146Łódź.

Proszę piszcie!!
Z poważaniem
Krzysztof Jasiński

1988.09.13

Mam ATARI 800 XL. Potrzebuję pomocy w grze CHIMERA i MONTEZUMA'S — nie mogę przejść komnaty z wieloma kładkami. Szukam też instrukcji do SILENT SERVICE. W zamian proponuję wymianę gier i doświadczeń.
Radosław Mielczarek
87-408 Ciechocin Dom Nauczyciela

Poszukuję gry THE GOONIES w wersji na Atari 65 XE. W zamian oferuję różne gry, a także opis BRUCE LEE.
Sebastian Streich ul. Pińska 44 03-523 Warszawa

Szukam opisów do gier B-1 NUCLEAR BOMBER i VIETCONG. W zamian opis do STARQUAKE.
Rafał Roman ul. Ugorek 10 c/14 31-456 Kraków

Mam około 50 gier na Atari 800 XL. Jednak nie umiem sobie poradzić z wieloma grami, w tym z grą PITFALL. Liczę na pomoc czytelników.
Mateusz Pilich ul. Kopernika 6 m 88 05-800 Pruszków

Proszę o pomoc w grach: PITFALL II, SHABOL WORD jak również szukam gier DIG-DUG, KARATEKA, REVERSI I i II.
Błażej Markiewicz ul. Obr. Wybrzeża 8 B m 98
80-398 Gdańsk

Potrzebny jest mi opis do gier ZORRO i CHIMERA.
Igor Przyjemski ul. Inowrocławska 14 m 8 91-033 Łódź

Interesuję się grami komputerowymi. Najbardziej podoba mi się SABOTEUR, ale nie mogę jej nigdzie dostać. Proszę o informację na ten temat.
Krzysztof Słupianek ul. Janasa 3a m 34
40-855 Katowice

Atari — proszę o nieśmiertelności do gier RIVER RAID i DROPZONE.
Adam Staszak ul. Głogowska 259 60-111 Poznań

Bardzo proszę o przysłanie mi schematów mikrokomputerów ZX Spectrum i ZX 81 lub informację, gdzie można zdobyć takie schematy.
Grzegorz Witek Czaniec 975
34-314 Czaniec woj. bielskie

Poszukuję następujących gier: KUNG-FU MASTER, YIEAR KUNG-FU, RAMBO, PHOENIX WINTER GAMES. Komputer Atari.
Robert Kałmucki os. Widok 16 m 18
66-200 Świebodzin woj. zielonogórskie

Nie wiem, jak uruchomić SPITFIRE 40- SKY FOX i GREEN BERET na Spectrum. Kto mi w tym pomoże?
Wojciech Nietupski ul. Chałubińskiego 22/11
80-807 Gdańsk — Chełm

Mam grę THE HULK (tekstową), ale nie mogę sobie dać z nią rady. Nie wiem co robić po zniszczeniu krzesła. Szukam tez opisu do LORD OF THE RINGS.
Marek Pecyna ul. Kolejowa 8 m 17 62-510 Konin

Od dwóch miesięcy trudzę się z zadaniami demona w grze TRAP DOOR. Udało mi się tylko oddać demonowi puszkę i zasadzi oczniaki. Jak wykonać pozostałe zadania?
Borys Suchodolski ul. Sikorskiego 44 m 27
40-282 Katowice

20 gier w zamian za: JACK THE NIPPER, BOMB JACK, PYJAMARAMA. COBRA STALLONE, POP EYE, KUNG-FU MASTER, RAMBO, BURCE LEE. Komputer C-64.
Tomasz Jajecznica ul. Garncarska 30 34-410 Rabka

Poszukuję opisu do gry EXCALIBUR na Atari.
Daniel Walesiak ul. Kościuszki 95 m 2 10-554 Olsztyn

Proszę o nieśmiertelności do gier: PYJAMARAMA, EVERY ONE'S WALLY, COMMANDO, GREEN BERET, POP EYE, JACK THE NIPPER. Szukam też gier: ALIENS, GLADIA TOR. SABOTEUR II, EXOLON. W zamian 20 innych. Jak skończyć gry: ASTERIX, RAMBO, POP EYE?
Jarosław Słowiński ul. Grodziska 30 60-363 Poznań

Jestem posiadaczem Amstrada CPC 464. Proszę o nieśmiertelność do gier: COMMANDO, KNIGHT LORE, BRUCE LEE, FRUITY FRANK,
Bartłomiej Habas ul. Chrobrego 17 m 111
40-881 Katowice

Proszę o pomoc we wpisaniu i uruchomieniu gry WINTER OLYMPIC w wersji na Atari 130 XE.
Michał Szczepaniak ul. Domowa 8 95-012 Łódź

Mam ZX Spectrum 48. Proszę o opisy do gier: COMMANDO, TRANSFORMERS.
Remigiusz Kola ul. Gdyńska 33 a 72-600 Świnoujście

Szukam opisów do gier na Atari 65 XE: GHOST BUSTERS, KING OF THE RING, FORT APOCALYPSE oraz kodu i opisu do gry THEATRE EUROPE.
Krzysztof Kucharz ul. Długa 41 m 13 41-800 Zabrze

Jak uzyskać nieśmiertelność w grze CRYSTAL RAIDER, a także jak uwolnić księżniczkę w grze ZORRO? Komputer Atari 65 XE.
Michał Pieprzyca ul. Niepodległości 56 a
41-404 Mysłowice

REKONFIGURACJA KLAWIATURY

Na 8-bitowych Amstradach do zmiany znaków lub ich ciągów przypisanych do poszczególnych klawiszy służy zbiór systemowy SETKEYS. COM. Jest on szczególnie użyteczny jeśli chcemy np. wprowadzić w wygodny sposób polskie litery na klawiaturze.

Kolejnym zastosowaniem może być przypisanie niektórym klawiszom słów kluczowych danego języka programowania. Przykładowo, naciśnięcie klawiszy EXTRA E generuje napis END, EXTRA B — BEGIN itp. Możliwe jest także zautomatyzowanie pewnych rutynowych czynności poprzez przypisanie poszczególnym klawiszom często stosowanych sekwencji: 'dir a: [size] <CR>','pip lst: = '.

Niestety program SETKEYS działa w sposób mało przyjemny dla użytkownika. Przed jego uruchomieniem trzeba przygotować zbiór tekstowy, w którym znajdują się linie opisujące wszystkie zmiany. Kolejną wadą jest to, że z programu tego nie możemy skorzystać będąc np. w interpreterze BASIC-a lub w swoim własnym programie.

Przedstawione poniżej dwa zbiory KEY.MAC i KEY.INC (rys. 1,2) umożliwiają rekonfigurację klawiatury w programach napisanych w języku Turbo-Pascal. Na rys. 3 podano przykładowy program, który przypisuje klawiszowi F1 na Amstradzie PCW napis Turbo' zakończony znakiem przejścia do początku linii — chr(13).

Zbiór KEY.MAC służy do generacji zbioru typu RSX, który dołączony do innego programu wynikowego pozwoli na korzystanie w nim z dwóch funkcji systemowych: KM SET EXPAND i KM SET KEY. Pierwsza z nich pozwala przypisać ciągi znaków w jednej z tablic systemowych, druga zaś przypisuje te ciągi lub pojedyncze znaki poszczególnym klawiszom. Te same funkcje realizowane są przez program SETKEYS, warto więc zapoznać się z jego opisem w instrukcji użytkownika komputera.

Do zapewnienia łączności RSX'a z programem w Pascalu służą dwie procedury (rys. 2) SetToken i SetKey. Zbiór KEY.INC zawierający je należy dołączyć do własnego programu przy pomocy opcji INCLUDE kompilatora Turbo-Pascala: (*$I KEY.INC*)

Procedura SetToken ma trzy argumenty:
1. TokenNo— numer ciągu znaków (liczba z zakresu 80-9F Hex),
2. StringToSet— ciąg znaków, które chcemy przypisać,
3. Error — zmienna logiczna, na którą zwracane jest TRUE, jeśli przypisanie zostało przyjęte przez system. Niekiedy zwłaszcza przy dużej ilości napisów, może nastąpić przepełnienie tablicy systemowej.

Procedura SetKey ma również trzy argumenty:
1. KeyNo — sprzętowy numer klawisza (patrz rysunek w instrukcji komputera),
2. Status — zmienna określająca czy klawisz został naciśnięty sam czy też razem z innym klawiszem typu: SHIFT, ALT, SHIFT Z ALT'em, EXTRA.
3. CharNo — kod znaku. Od 0 do 7F są to kody ASCII, od 80 do 9F ciągi znaków przypisywane funkcją SetToken, a od A0 do FC dalsze znaki

Generacja RSX'a odbywa się przy pomocy assemblera M80 i programu konsolidującego LINK. COM:

A>m80 =KEY.MAC
A>LINK KEY [OP]
A>REN KEY.RSX = KEY.PRL

Dołączenie RSX'a do programu wynikowego DEMOKEY.COM, uzyskanego z kompilatora Turbo-Pascala następuje przez działanie programu systemowego GEN - C0M.COM:

A>GENCOM DEMOKEY.COM KEY.RSX

Uwaga. Kompilację w Turbo-Pascalu należy przeprowadzić przy zmniejszonym adresie końcowym zbioru wynikowego (patrz poprzednie artykuły w Klanie Amstrada).

Jonasz Mayer

```
; ******************************
****
;   rsx realizujacy funkcje
;   systemowe PCW 8256 i CPC 6128:
;    1. KM SET EXPAND
;    2. KM SET KEY
;    (C) JK kwiecien 88
; ******************************
****
      .z80
token   equ 9*
key     equ 91
warm    equ 0001H
userf   equ 3*(30-1)
bufsize equ 80
;
 cseg ; rsx header
 ds 6
 jp start
bdos: jp 8000h
 dw 0
 db OFFH;
 db 0   ;
 db 'TokenKey'
 db 0,0,8
start:
 ld a,c
 cp token
 jp z,tkn
 cp key
 jp bdos
evadr:
 id hl,(warm)
 ld be,userf
 add hl,bc
 ld (adr),hl
 ret
jphl: db 8c3h
adr: ds 2 buffer:
 ds bufsize tkn: call evadr
 ex de,hl ; hl wsk Paramblock
 ld b,(hl) ; b=token
 inc hl
 ld c,(hl) ; c=dlugosc
 inc hl
 push bc ; zapamietaj b i c
 ld de,buffer
 push de
 ld b,0
 ldir  ; przeslanie do bufora
 pop hl ; odtworz adress bufora
 pop bc ; odtworz b i c
 call jphl ; wywolanie funkcji
 dw 00d4h ; KM SET EXPAND
 ld a,l
 ret c ; normalny powrot
 xor a
 ret  ; powrot z bledem
sk: call evadr
 ex de,hl ; hl wsk paramblock
 ld c,(hl) ; c = numer klawisza
 inc hl
 ld d,(hl) ; d= status
 inc hl
 ld b,(hl) ; b= numer znaku
 call jphl ; wywolanie funkcji
 dn 00d7h ; KM SET KEY
 ret
end
```

```
(*************************************************)
(* Zbior KEY,INC - rekonfiguracja klawiatury w   *)
(*    programach Turbo-Pascalowych przy          *)
(*         wykorzystaniu RSX'a                   *)
(*         (C) JM    kwiecien 1988               *)
(*************************************************)
type
    Status = (Normal,Shift,Alt,SAlt,Extra);
    AnyStr = String(,80,);
    Token  = $80..$9F;
procedure SetToken (TokenNo      : byte;
                    StringToSet  : AnyStr;
                    var Error    : boolean);
const
     RSXno = 90;
var
     ParamBlock : record
                    T : byte;
                    S : AnyStr;
              end;
begin
  if (TokenNo<$80) or (TokenNo>$9F)
  then Error := true
  else begin
          with Paramblock
          do begin
                  T := TokenNo ;
                  S := StringToSet;
          end;          Error :=
         (BDOS (RSXno,addr(Paramblock)) = 0);
      end;
end; (* of Set Token *)

procedure SetKey (KeyNo  : byte; State : Status;
                  CharNo : byte);
const
     RSXno = 91;
var
     ParamBlock : record
                    k, s, c : byte;
              end;
begin
     with ParamBlock
     do begin
             k := KeyNo;
             case State
             of  Normal :  s :=  1;
                 Shift  :  s :=  2;
                 Alt    :  s :=  4;
                 SAlt   :  s :=  8;
                 Extra  :  s := 16;
             end;
             c := CharNo;
      end;
   BDOS (RSXno,addr(Paramblock));
end; (* of Set Key *)

program DemoKey;
(*********************************************)
(* przyklad zastosowania funkcji systemowych *)
(* dostepnych w Turbo-Pascalu dzieki zbiorowi*)
(* KEY.INC                                   *)
(*********************************************)
(*$IKEY.INC*)
var
   Error : boolean;
begin
     SetToken ($80,'Turbo'+chr(13),Error);
     if error
     then write('Blad w procedurze Set Token')
     else write('OK. ALT-F1 uruchamia Turbo-Pascal');
     SetKey(2,Alt,$80);
end.
```

KATALOG DYSKIETKI

W TURBO-PASCALU

W SYSTEMIE CP/M

(KOMPUTERY: AMSTRAD PCW 8256 /8512, AMSTRAD CPC 6128, COMMODORE 128, itd)

Turbo-Pascal jest kompilatorem języka PASCAL dostępnym na komputerach pracujących pod jednym z dwóch systemów operacyjnych: CPI /M i MS DOS. O ile w przypadku MS DOS a mamy praktycznie jeden typ sprzętu — IBM PC lub kopie — to dla CP/M'u liczba różnych maszyn jest bardzo duża.

PWiele atrakcyjnych operacji, potrzebnych użytkownikowi zależnych jest od sprzętu. Dotyczy to takich zagadnień jak: obsługa ekranu w trybie graficznym, dostęp do zegara lub katalogu dyskietki. Dla komputerów kompatybilnych z IBM PC oferowane są dodatkowe pakiety programów Turbo-Pascalu np. Graphics Toolbox, Power Utilities i inne. Oprogramowanie dostępne w systemie CP/M jest znacznie uboższe i z powodu różnorodności sprzętu znacznie mniej specjalizowane. Dla większości posiadaczy 8-bitowych komputerów koniecznością jest samodzielne tworzenie potrzebnego oprogramowania. W przypadku wykorzystywania charakterystycznych cech danej maszyny wymagana jest dobra znajomość systemu operacyjnego i jego konkretnej implementacji

Jedną z bardziej pożądanych możliwości przy pisaniu oprogramowania, nie tylko profesjonalnego, jest umiejętność czytania katalogu dyskietki.

Przedstawiony poniżej program zawiera procedurę DIR, która wypisuje zbiory znajdujące się na dyskietce i liczbę wolnych kilobajtów.

Pierwszą czynnością przy odczycie katalogu jest zainicjowanie tzw. FCB (ang. File Control Block), czyli tablicy kontrolnej pliku. Jest to wektor bajtów, którego 12 pierwszych miejsc musi być wypełnionych przed odwołaniem się do systemu. Znaczenie tych pól jest następujące:

FCB (0) : numer stacji dysków: (0 — aktualnie dostępna. 1 — stacja A, 2 — stacja B, itd.)

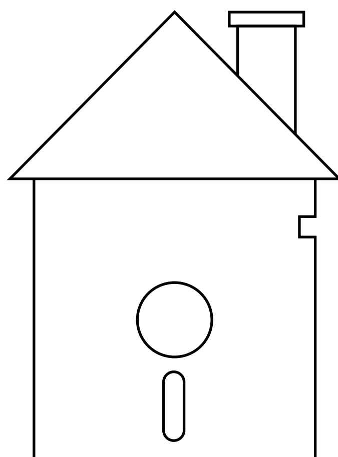
FCB (1) — FCB (8): nazwa zbioru (8 znaków alfanumerycznych), jeśli chcemy uwzględnić wszystkie zbiory, to wstawiamy w każde miejsce "?"

FCB (9) — FCB (11): rozszerzenie (3 znaki), znak „?" ma to samo. co poprzednio, znaczenie.

Znalezienie pierwszego zbioru w katalogu dyskietki pasującego do podanej nazwy i rozszerzenia odbywa się przez zastosowanie funkcji BDOS'a a numerze 17 (Search First). Drugim argumentem tej funkcji oprócz numeru jest adres FCB. Następne wystąpienia nazwy zbioru znajdowane są przy pomocy osiemnastej funkcji BDOS'a (Search Next).

Do znalezienia wielkości wolnego obszaru na dysku służy 46 funkcja BDOS'a (Free Space). Z tej funkcji można korzystać dopiero w CP/M Plus. Jedynym argumentem procedury DIR jest numer stacji dysków. Poprzez dodanie dalszych parametrów i skorzystanie z dostępnych w TURBO-Pascalu funkcji np FileSize można napisać własną procedurę zastępująca działanie programów systemowych takich jak STAT.COM lub DIR.COM,

Jarosław Młodzki



```
program CPM80DirDemo;
(************************************)
(* program demonstracji procedury DIR *)
(* sporzadzajacej katalog dyskietki   *)
(************************************)
var
     DriveNo : byte;

procedure Dir ( No : byte );
(*****************************************************)
(* wypisuje zbiory na dyskietce  i liczb  nie zaj tych  *)
(* kB - zmienna Free                                  *)
(* argumentem procedury jest numer stacji             *)
(* No = 1 stacja A:                                   *)
(*      2 stacja B:                                   *)
(*     13 stacja M; (RAMDYSK)                         *)
(*****************************************************)
const
     Search_First    : integer = $11;
     Search_Next     : integer = $12;
     Set.DMA         : integer = $1A;
     FreeSpace       : integer = $2E;
     DefaultDMA      ; integer = $0080;
  var
     Error, i, Start,
     Free, Counter   : integer;
     FCB             : array(.0..25.) of Byte absolute $005C;
     DMA             : array(.0..255.) of Byte;
begin
     Error := BDos(Set_DMA,Addr(DMA));        (* ustaw nowe DMA *)
     FCB(.0.) := No;                  (* ustaw numer stacji     *)
     for i := 1 to 8                  (* ustaw maske dla nazwy  *)
     do FCBC(.i.) := Ord('?');
     for i := 9 to 11           (* ustaw maske dla rozszerzenia *)
     do FCBC(.i.) := Ord('?');  (* znajdz pierwszy zbi r        *)
     Error := BDos(Search_First,Addr(FC6));
     if Error<>255
     then begin
             Start := Error * 32;
             for i := Start to Start+8
             do write(Char(Mem(. Addr(DMA)+i.)));  write(' '),
             for i := Start+9 to Start+11
             do write(Char(Mem(.Addr(DMA)+i,)));   write(' ',8);
        end;
     counter := 1;                    (* znajdz nastepne zbiory *)
     repeat
            Error := BDos(Search_Next);
            Start := Error * 32;
            if Error<>255
            then begin
                      for i := Start to Start+8
                      do write(Char(Mem(.Addr(DHA)+i.))); write(' ');
                      for i := Start+9 to Start+11
                      do write(Char(Mem(.Addr(DMA)+i.))); write(' ',8);
                      counter := counter + 1;
                      if counter = 4
                      then begin
                                counter := 0; writeln;
                           end;
                 end;
            until Error=255;
            if counter<>9 then writeln;    (* wolne miejsce na dysku *)
            BDOS (Set_DMA,DsfaultDMA);     (* przywroc zwykle DMA    *)
            if (No>0) and (No<15)
            then begin
                     BDOS (FreeSpace, No-1);
                     Free := (Mem(.DefaultDMA,)+256*Mem(.DefaultDMA+1,)) div 8;
                     writeln('wolne ',Free, 'kB');
                 end;
end; (* of dir *)
begin
      write (' podaj numer stacji: ( 1 - A;  2 - B;  13 - M; ) ');
      readln (DriveNo);
      writeln;
      Dir (DriveNo);
end.
```

W KOLEJCE DO



O tym, że komputer jest przede wszystkim narzędziem pracy nie zaś zabawką pisaliśmy w "Bajtku" już nie raz. Dziś tropiąc poważne zastosowania komputerów spenetrujemy wnętrze nowoczesnego sklepu.

Jeśli sklep jest pełen towarów, to zakupy mogą być dużą przyjemnością. Zawsze jednak pozostaje moment, który przyjemny nie jest. Oczywiście mam na myśli płacenie. Nie dość, że musimy uszczuplić zawartość portfela, to często tracimy przy tej okazji sporo czasu. Zależy to trochę od techniki pracy obsługi i w znacznej części od urządzeń, którymi ta obsługa dysponuje. Zdarza się, że ekspedientka sumuje należności w pamięci lub zapisuje i dodaje kwoty na papierku. Czasami używa kalkulatora a w części sklepów zainstalowane są nawet specjalne duże kasy, tzw. kasy rejestracyjne. Usprawniają one pracę kasjerki, gdyż po wystukaniu na klawiszach należności za wszystkie towary potrafią podsumować rachunek i wydrukować paragon dla klienta. W tej chwili w naszych sklepach można zwykle spotkać kasy elektromechaniczne — zarówno obliczenia jak też ustawianie wyniku do odczytu przez człowieka odbywają się dzięki współpracy odpowiednich trybików i przekaźników, poruszających się wewnątrz urządzenia. Taka kasa może być jednak wygodniejsza w użyciu niż kalkulator, gdyż jest odpowiednio wyspecjalizowana do pracy w sklepie. Np. może, oprócz sumowania rachunków dla poszczególnych klientów, zsumować cały dzienny utarg. Takich dodatkowych, bardzo przydatnych dla obsługi sklepu funkcji możemy znaleźć dużo więcej. Powiemy o nich za chwilę, gdyż teraz chcemy wykonać śmiały krok w nowoczesność. A przy okazji tego kroku wyjaśni się również dlaczego piszemy o sklepie i kasach sklepowych w miesięczniku komputerowym.

Pierwszy objaw nowoczesności to oczywiście zastąpienie elementów mechanicznych elementami elektronicznymi. Wewnątrz dzisiejszych kas (prezentowanych na zdjęciach ilustrujących ten artykuł) siedzi elektronika analogiczna (a często identyczna) do elektroniki wypełniającej wnętrze mikrokomputera. Zwykle znajdziemy tam mikroprocesor i pamięć. Często zdarza się, że tej pamięci jest znacznie więcej niż w domowym mikrokomputerze, takim jak Commodore C-64. czy Spectrum. Niektóre kasy mają np. po 512 KB pamięci, rozbudowaną logikę, program działania zapisany w pamięci ROM, kilka złącz (najczęściej są to złącza RS-232) umożliwiających wszechstronną komunikację ze światem zewnętrznym. Do komunikacji z człowiekiem przeznaczone są wyświetlacze, zarówno cyfrowe jak i alfanumeryczne, pozwalające wyświetlać nazwy towarów. Do wydruku paragonów, rachunków i rejestracji operacji do celów kontrolnych mamy mini drukarki znakowo-mozaikowe (zwykle od jednej do trzech w zależności od przeznaczenia i konstrukcji kasy).

Czym trudni się góra elektroniki wypełniająca wnętrze kasy rejestracyjnej? Zacznijmy od najprostszych, zupełnie podstawowych funkcji. Oczywiście pozwala zarejestrować każdą transakcję, podsumowuje rachunki dla klientów, może obliczać wyświetlać ile należy wydać reszty. Podobnie jak w kasie tradycyjnej informacja o sprzedaży nie ginie wraz z odejściem klienta od kasy, lecz jest pieczołowicie przechowywana. Konkretnie, wszystkie transakcje zapisywane są w pamięci kasy, w razie potrzeby wraz z pewnymi dodatkowymi danymi, np. godziną dokonania, formą zapłaty: czek/gotówka, itd. Rejestracja czasu sprzedaży wymaga umieszczenia w kasie zegara, ale przecież to żadna trudność. Idźmy dalej. Kasę może obsługiwać na zmianę kilku kasjerów. Przed rozpoczęciem pracy każdy musi podać swój kod a kasa oddzielnie będzie zapamiętywać utarg każdego z nich — ułatwi to kierownikowi rozliczenie.

Jeśli sklep jest dzielony na odrębne stoiska, to przy wpisywaniu każdego z towarów kasjerka może dodatkowo wcisnąć klawisz oznaczający stoisko. Kasa będzie zliczać utarg dla każdego stoiska oddzielnie. Analogicznie można podzielić towary na pewne grupy i oddzielnie podliczać sprzedaż towarów każdej z grup.

Jak widać na koniec dnia praca kasy pełna jest nie tylko pieniędzy, a również danych o tym co działo się przez cały dzień. Drukarka, która normalnie drukowała paragony posłuży teraz do wydruku szczegółowego raportu.

Aby pieniądze były bezpieczne trzymamy je w dołączanej do kasy specjalnej szufladzie. Jeśli chodzi o zabezpieczenie danych szuflada nie pomoże wiele. Aby nie zniszczył ich przypadkowy spadek napięcia w sieci kasa ma własne awaryjne zasilanie z baterii.

Spójrzmy teraz na handel od drugiej strony. Przez zaplecza i magazyn płynie strumień towarów. Aby na sklepowych półkach nie zabrakło, np. krawatów tylko dlatego, że



KASY

ktoś nie pomyślał w porę o ich zamówieniu, cały ruch towarów musi być szczegółowo śledzony i analizowany. Dla dużych sklepów ewidencja towarów jest naprawdę ogromną pracą więc jak zwykle w takich przypadkach ludzie zwalili robotę na komputery. Do komputera wprowadzamy dane o tym co zostało dostar czone z fabryki czy hurtowni. Stan towarów musi być jednak aktualizowany w miarę ich sprzedaży. Od razu nasuwa się łatwe i oszczędzające pracy rozwiązanie: niech kasa dane o sprzedaży każdej sztuki przesyła bezpośrednio do komputera. Pożytki są oczywiste — można na bieżąco śledzić stany towarów, można wyeliminować wiele zbędnej papierkowej roboty, czyli zaoszczędzić czas i pieniądze.

Pojawiają się jednak pewne trudności techniczne. Do tej pory kasjerka wystukiwała cenę każdego towaru, a przecież towarów o tej samej cenie może być kilka. Skąd komputer będzie wiedział co zostało sprzedane? Aby temu zaradzić trzeba wprowadzić pewien kod, czyli sposób oznaczania towarów. Każdy towar musi mieć własny symbol, który będzie go jednoznacznie identyfikował. Spotykamy tu różne warianty rozwiązania. Kod może np. być literowy lub cyfrowy, kasjerka odczyta go z towaru i wpisze na klawiaturze, tak samo jak robiła to poprzednio z ceną. Natomiast inteligentny system może już oszczędzić wpisywania ceny. Towar o wpisanym kodzie jest odszukany w pamięci i stamtąd pobieramy jego cenę. Następnie fakt dokonania transakcji zostaje zapisany w pamięci. Powstaje tu naturalne pytanie: w czyjej pamięci, kasy czy komputera? Tu znowu możliwe są dwa podstawowe rozwiązania (i oczywiście wiele wariantów pośrednich). Albo cała lista towarów znajduje się w pamięci kasy i tam też odnotowywane są wszystkie transakcje. Wtedy kasa pracuje w pełni autonomicznie (ang. off-line) i jest niezależna od komputera, a więc w szczególności niewrażliwa na jego awarie. Połączenie z komputerem następuje od czasu do czasu — komputer przesyła wtedy informacje o zmianach na liście towarów, np. o przecenach lub nowych dostawach, pobiera zaś informacje o tym co zostało sprzedane. Drugi wariant to współpraca kasy z komputerem przez cały czas — wymiana danych ma miejsce przy sprzedaży każdej sztuki. W tej organizacji kasa (która pracuje on-line) jest właściwie inteligentną, specjalistyczną komórką systemu komputerowego.

Wróćmy teraz do kodowania. Zamiast liter czy cyfr można do oznaczania towarów użyć kombinacji kresek. Powstaje wtedy kod określany jako kod kreskowy lub kod prążkowy (ang. bar code), Zobaczyć go możemy na wielu towarach trafiających do nas z krajów zachodnich, gdzie ten sposób kodowania jest bardzo rozpowszechniony. Zaletą tej metody jest możliwość czytania kodu przez specjalny czytnik podłączony do kasy. Likwiduje to konieczność czytania oznaczeń przez kasjerkę i pracochłonne wystukiwanie na klawiaturze. Czytnik przesyła kod towaru do kasy i dalej proces wygląda tak samo jak w poprzednim przypadku.

Czytniki kodu kreskowego produkowane są w najróżniejszych postaciach: ołówka lub pistoletu, którym trzeba przesunąć po opakowaniu, czy też lady, nad którą wystarczy machnąć w powietrzu oznakowanym towarem.

Zwykle już producent umieszcza odpowiedni kod na opakowaniu towaru. Jeśli nie zrobił tego, lub jeśli ten kod nie odpowiada naszym potrzebom to możemy sobie poradzić sami. Wystarczy kupić odpowiednią drukarkę i program, który pozwoli tworzyć własny kod (potrzebny będzie też komputer, ale mieliśmy go już przedtem). Do tego zamiast papieru do drukarki folia samoprzylepna, oczywiście na podkładzie z perforacją na krawędziach, pozwalającą wygodnie zakładać ją do drukarki. Taki zestaw pozwoli nam samodzielnie tworzyć oznakowanie towarów kodem kreskowym. Wydrukowane metki z kodem nakleimy na towary, najlepiej za pomocą specjalnego pistoletu metkującego.

Mamy więc już ułożony podstawowy schemat przebiegu danych w nowoczesnym sklepie. Jak widać jest on oparty przede wszystkim o nowoczesne rozwiązania elektroniczne — komputery i kasy rejestracyjne.

Spróbujmy dorzucić jeszcze kilka szczegółów. Jeśli sklep sprzedaje towary wymagające ważenia, to wagę (oczywiście też elektroniczną) warto podłączyć bezpośrednio do kasy. Jeśli wielu naszych klientów ma magnetyczne karty kredytowe lub debetowe, to warto wyposażyć kasy w czytniki takich kart. To zaś z kolei może wymagać podłączenia kasy do jeszcze jednej sieci — komputerowej sieci dostępu do danych bankowych.

Zgodzicie się chyba, że żeby móc wykonać wszystkie swoje zadania elektroniczne kasa musi mieć w środku całkiem niezły komputer. Oczywiście nie wszystkie kasy mają tyle rzeczy do zrobienia. Są więc również modele prostsze, o znacznie mniejszych możliwościach i ... cenie. To ostatnie też chyba istotne, bo najbardziej skomplikowane kasy kosztują po kilka tysięcy dolarów, a więc więcej niż tajwańskie kopie IBM PC.

Ponieważ zbliżamy się już do końca, wyjdźmy ze sklepu. Nie musi to jednak wcale oznaczać rozstania się z kasami rejestracyjnymi. Ich odpowiednie wersje instaluje się przecież w restauracjach, hotelach, na stacjach benzynowych. Wszędzie gdzie mogą ułatwić i przyspieszyć obsługę klienta oraz usprawnić zarządzanie.

Andrzej Pilaszek

PROGRAMOWAĆ MOŻE KAŻDY

KONTROLOWANIE JOYSTICKA Z BASIC-u,

czyli nie jest tak źle, jak myślicie

Pośród różnych ograniczeń wynikających z pisania programów w Basic-u jedno, związane z szybkością działania programów, jest szczególnie bolesne. Myślę tu o używaniu joysticka. Wyobraźmy sobie taką sytuację — piszemy grę, w której należy sterować poruszającym się po ekranie obiektem. Najlepiej by było robić to przy pomocy joysticka. Niestety, istniejąca w wielu mikrokomputerach możliwość odczytania w dowolnym momencie stanu portu, do którego joystick jest podłączony, nie daje się w praktyce wykorzystać, gdyż wyliczenie wartości poszczególnych bitów w odczytanym bajcie jest w Basic-u tak czasochłonne, że po kilku próbach wszyscy decydują się na sterowanie przy pomocy klawiatury. Przez długi czas ja również postępowałem w ten sposób, aż w końcu przyszło olśnienie, którego efektem jest poniższy program.

Najważniejszą jego częścią jest podprogram zawarty w liniach od 20 do 70. Podprogram ten wykonuje następujące czynności — w linii 20 zostają wyzerowane zmienne L, S, G i D, które odpowiadają pochyleniu joysticka odpowiednio w Lewo, w Górę, w Dół i naciśnięciu przycisku Strzał, a następnie zmiennej P (odpowiadającej pochyleniu joysticka w Prawo) zostaje nadana wartość odczytana z portu 31 (co na Spectrum odpowiada interfejsowi Kempston). W liniach od 30 do 60 wartości zmiennych są w zależności od stanu joysticka pozostawione w spokoju lub zamienione na 1 (co odpowiada sygnałowi „joystick pochylony w tą stronęł”). W lini 80 następuje zainicjowanie zmiennych X i Y, czyli położenia obiektu którego ruchem będziemy sterować. W liniach od 90 do 130 znajduje się główna pętla naszego programu, w której drukowany jest poruszany obiekt (jego kolor zależy od tego czy guzik „strzał” jest naciśnięty czy zwolniony), i obliczane są jego nowe współrzędne.

Jeszcze kilka słów wyjaśnienia na temat podprogramu z linii 20. Nie będę tłumaczyć algorytmu, natomiast radzę prześledzić jego działanie dla różnych wartości początkowych zmiennej P. Efektem jego działania jest nadanie zmiennym L, S, G. P i D wartości 0 lub 1, odpowiadających wartościom poszczególnych bitów w bajcie odczytanym z portu 31. (Znających asemblera Z-80 zaciekawi pewnie informacja, że na zastosowane rozwiązanie naprowadziły mnie instrukcje SRL i SLA.) Przez zastąpienie liczby 31 w lini 20 inną, możemy dostosować go do innego interfejsu niż Kempston, pod warunkiem zmienienia również kolejności nadawania wartości zmiennym L, S, G. P i D (tak, żeby odpowiadały odpowiednim bitom bajtu określającego stan joysticka). Przypuszczam, że po zmienieniu funkcji IN w lini 20 na jej odpowiednik w innym dialekcie Basic-u, podprogram może być użyteczny nie tylko dla użytkowników Spectrum. Można go również zastosować wszędzie tam, gdzie potrzebne jest przedstawienie liczb w postaci binarnej, jeśli używany dialekt Basic-u nie ma odpowiedniej instrukcji.

Marcin Borkowski

```
10 GO TO 80
20 LET L=0:LET S=0:LET G=0:LET D=0:LET P=IN 31
30 IF P>-16 THEN LET S=1:  LET P-P-16
40 IF P>=8 THEN LET G=1:  LET P=P-8
50 IF P>=4 THEN LET D=1:  LET P-P-4
60 IF P>=2 THEN LET L-1:  LET P=P-2
70 RETURN
80 LET X=16:  LET Y-10
90 GO SUB 20
100 PRINT AT Y,X; PAPER 5-S;" "
110 LET Y=Y-G+D
120 LET X=X+P-L
130 GO TO 90
```

WSZYSTKO DLA WSZYSTKICH

GIEŁDA 17.09.88

	GIEŁDA BAJTKA (tys. zl)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
SINCLAIR			
ZX Spectrum 48KB	125	115	90
ZX Spectrum Plus	160	–	99
ZX Spectrum + 2	260	–	180
ZX Spectrum + 3	325	–	280
TIMEX 2048	165	146	–
COMMODORE			
C-64C	300	219	299
C-128	400	299	399
C-128 D	900	–	777
Magnetofon 1530, 1531	70	48	40
Stacja dyskietek 1541	310	–	315
Stacja dyskietek 1570	345	–	369
Stacja dyskietek 1571	360	299	495
Drukarka MPS 801	210	–	–
Drukarka MPS 803	270	–	250
AMIGA 500	1,4	–	888
C-16	120	–	90
ATARI			
65 XE	170	125	100
130 XE	200	199	220
Stacja dyskietek 1050	260	185	299
LDW 2000 Super	360	199	–
XC-12	60	48	48
AMSTRAD			
464 mono monitor	650	–	350
464 kolor monit.	725	–	450
6128 mono monit.	895	–	689
6128 kol. monit	1.0	–	777
PC 1512 SD MM	1.5	–	889
Dyskietki 3"	5	–	4-9
Dyskietki 3'.5"	2.8	–	3-9
Dyskietki 5.25"	0.9–1.5	–	0.6
Joystick	6–15	4–9	6–25

Sklep Bajtka w BYTOMIU ul. Koniewa 6 tel. 81-57-01	
ZX-Spectrum 48 K	130.000
ZX-Spectrum plus	175.000
ZX-Spectrum +2	280.000
Seikosha GP-50S	120.000
Commodore 64	240–270.000
Commodore 128	380.000
Commodore 128D	–
AMIGA 500	–
Commodore 16	120.000
Commodore 116	100.000
Commodore +4	165.000
Magnetofon 1530	50– 60.000
Stacja dyskietek 1541	270.000
Atari 65 XE	190.000
Atari 130 XE	260–290.000
XC-12	60-65.000
Stacja dysków 1050	240–270.000
LDW 2000 Super	270.000
464 mono	340.000

INDYWIDUALNY BANK DANYCH

Tadeusz Kiernicki, lat 27. Posiada Atari 65 XE, magnetofon CX 12, stację dysków 1050. Jest inwalidą I grupy mieszkającym na wsi. Prosi o kontakt z innymi posiadaczami tego komputera. Adres: 24-132 Stary Pożógh 2, woj. lubelskie

Ryszard Trela, posiada komputer SHARP MZ-800 oraz bardzo dużo programów na wersję MZ-700. Pragnie wymienić się programami i doświadczeniami, poszukuje polskich instrukcji do programów użytkowych, języków programowania Adres: 42-500 Będzin, ul. Walki Młodych 3/19.

Jacek Trojak, lat 24. Posiada mikrokomputer ZX Spectrum 48 kb. Oprogramowanie: gry i programy użytkowe. Szczególnie zainteresowany jest programami edukacyjnymi. Adres: 43-300 Bielsko-Biała, ul. Lenartowicza 8/11.

Marek Rachel, lat 14. Posiada mikrokomputer Atari 65 XE, magnetofon XC-12, monitor monochromatyczny. Proponuje wymianę oprogramowania. Adres: 63-600 Kępno, ul. Wileńska 4.

Dariusz Orłowski, lat 16. Posiada mikrokomputer MSX Spectravideo 738, magnetofon, monitor oraz stację dysków. Proponuje wymianę doświadczeń. Adres: 164 75 Kista, Sairagatan 9. Szwecja.

Tadeusz Końpa, lat 17. Posiada mikrokomputer ZX Spectrum 48 kb, magnetofon, monitor oraz joystick. Oprogramowanie: około 60 gier, programy użytkowe. Proponuje wymia nę doświadczeń i oprogramowania. Adres: 63-101 Śrem, ul. Zawadzkiego 19/14.

Andrzej Nazdraczew, uczeń lat 17. Posiada mikrokomputer TIMEX 2068 oraz firmowy magnetofon. Zainteresowania akwarystyka oraz wędkarstwo. Proponuje wymianę programów. Adres: 71-540 Szczecin, ul. Czcibora 10/10.

Wojciech Jaśkiewicz, lat 22. Komputer Spectrum +. Posiada około 1000 programów w tym 700 gier. Nawiąże kontakt w celu wymiany oprogramowania. Adres: 59-200 Legnica, ul. Lotnicza 36/5.

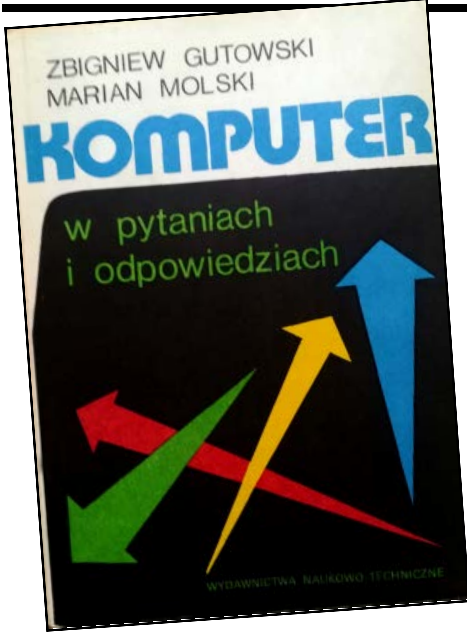
Wiktor Marysiewicz, lat 17. Posiada mikrokomputer CPC 464 z monitorem monochromatycznym. Pragnie nawiązać kontakt w celu wymiany gier i programów użytkowych. Adres: 59-225 Chojnów ul. Grodzka 2.

Sebastian Cieślak lat 13. Posiada mikrokomputer ATARI 800 XL oraz magnetofon XC 12. Pragnie nawiązać kontakt z użytkownikami ATARI, w celu wymiany doświadczeń, oprogramowania oraz literatury. Adres: 72-200 Nowogard ul. Bohaterów, Warszawa 80.

Jacek Osadziński, lat 40 — nauczyciel. Posiada TIMEX 2048 oraz okoto 100 programów (głównie użytkowe i edukacyjne oraz kilka własnych). Zainteresowania: informatyka, muzyka, matematyka, fizyka, informatyka. Proponuje wymianę programów i doświadczeń. Poszukuje programu do układania planu lekcji w szkole. Adres: 63-014 Murzynowo Kościelne.

Wiesław Pawłowski lat 33. Mikrokomputer ATARI 65 XE, stacja dysków 1050, magnetofon XC 12. Oprogramowanie: programy użytkowe oraz gry. Nawiąże kontakt z użytkownikami tego mikrokomputera, proponuje wymianę doświadczeń oraz oprogramowania.

Tomasz Aniszewski, uczeń lat 18. Posiada AMSTRADA PC 1512 HD20 MM oraz bogate oprogramowanie. Proponuje wymianę doświadczeń, oprogramowania i literatury. Adres: 04-854 Warszawa, ul. Króla Kazimierza 12.



Czy książka o komputerach długo będzie oczekiwać na półkach księgarni zanim znajdzie czytelnika i czyjej sprzedaż przyniesie odpowiedni zysk? Pytania te zadają sobie wydawcy przed podjęciem decyzji o produkcji każdej nowej pozycji. Hobbystów miłośników techniki komputerowej nurtuje z kolei problem, skąd czerpać informacje potrzebne do pogłębiania swoich zainteresowań.

Oczekiwania jednych i drugich udało się szczęśliwie spełnić autorom książek „Mikroprocesor w pytaniach i odpowiedziach” oraz „Układy scalone w pytaniach i odpowiedziach". Być może właśnie duża poczytnośc dwóch wydań każdej z nich zachęciła Wydawnictwa Naukowo-Techniczne do kontynuowania serii. W każdym razie mnie osobiście bardzo ucieszyło, że mogłem uzupełnić swój zbiór o „Komputer w pytaniach i odpowiedziach”.

Książka ta w połączeniu z poprzedniczkami tworzy pewną całość pozwalającą poznać komputer ze wszystkich stron. Mikroprocesor i układy scalone, które są podstawowymi cegiełkami służącymi do budowy wszelkich urządzeń elektronicznych, nie będą już ukrywać wielu tajemnic po uważnym przeczytaniu tej udanej serii. Dociekliwi czytelnicy, którzy stale poszukują nowych źródeł poszerzania swojej wiedzy, znajdą tu odpowiedź na pytanie, co mogą zdziałać scalak: zamknięte pod obudową komputera. Nie dziwmy się, że obok informacji bardzo szczegółowo ukazujących strukturę komputera natkniemy się tu także na tak proste pojęcia jak bajt, bit, procesor, program czy dane. Nie zapominajmy przy tym, że książka ta ma zaspokoić ciekawość nie tylko doświadczonych informatyków. Ma ona umożliwić nawet laikowi zdobycie potrzebnych mu elementarnych wiadomości. Pytania stawiane przez autorów pogrupowane są tematycznie. Każda ich grupa dotyczy pewnego szerszego zagadnienia. Tak więc obok hasła „procesor” umieszczono wszystkie najważniejsze związane z nim pojęcia: rejestry, licznik rozkazów, operacje logiczne, sposoby przedstawiania liczb itp. Do najbardziej pasjonujących fragmentów zaliczyłbym rozdział poświęcony pamięciom. Duża szybkość i niezawodność zapisu informacji jest jedną z najważniejszych cech charakteryzujących nowoczesny sprzęt komputerowy. Autorzy przedstawiają najciekawsze dotychczas opracowane rozwiązania techniczne umożliwiające przechowywanie danych: pamięci półprzewodnikowe, rdzeniowe i pamięci zewnętrzne, ze szczególnym uwzględnieniem różnych rodzajów pamięci dyskowych. Opisano tu między innymi szczegóły konstrukcyjne dysku twardego. Książkę warto przeczytać również ze względu na sporą porcję wiadomości o oprogramowaniu komputera i systemów komputerowych. Obszerną część poświęcono językom programowania, podstawowym funkcjom, które musi spełniać każdy program, sposobom jego uruchamiania oraz testowania. Przyda się to szczególnie wszystkim tym, którzy w przyszłości zamierzają sami zająć się tworzeniem nowego software. Na zakończenie autorzy zamieszczają krótki rys historyczny rozwoju komputerów i snują rozważania dotyczące tendencji dominujących dziś oraz przewidywania, czego możemy spodziewać się w przyszłości.

Forma pytań i odpowiedzi, którą posłużono się w omawianej książce, jest szczególnie atrakcyjna ze względu na łatwość dotarcia do żądanej informacji. Ułatwi to posługiwanie się innymi fachowymi opracowaniami, gdy jakiś trudniejszy termin wyda się nam niejasny.

(J.J.)

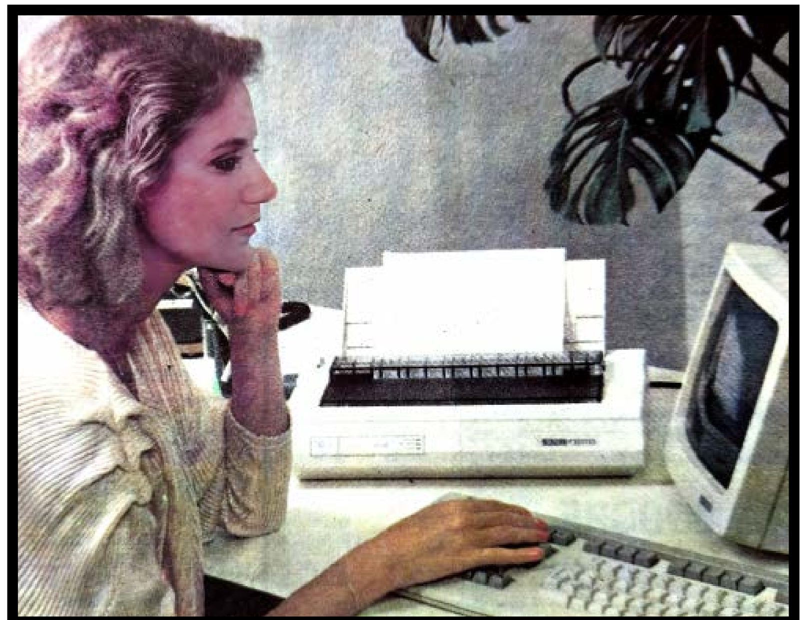
Zbigniew Gutowski, Marian Molski, „Komputer w pytaniach i odpowiedziach'', Wydawnictwa Naukowo- Techniczne, Warszawa 1988, Wydanie 1 Nakład 50.000 egz., Cena 680 zł.

NAJWIĘKSZY DYSTRYBUTOR PERYFERII KOMPUTEROWYCH W RFN

SYNELEC

Datensysteme
GmbH
Postfach 151727
8000 Munchen 15
tlx 5212289
tel 089/519278

CITIZEN



oferuje:

A) DRUKARKI MOZAIKOWE CITIZEN

	Cena/szt. DM	Transport DM
LSP-120D (9 igieł, 10 cali, 120 Zn/s)	350	40
MSP-15E (9 igieł, 15 cali, 160 Zn/s)	675	40
MSP-40 (9 igieł, 10 cali, 200 Zn/s)	715	40
MSP-45 (9 igieł, 15 cali, 200 Zn/s)	895	40
MSP-50 (9 igieł, 10 cali, 300/250 Zn/s)	1190	40
MSP-55 (9 igieł, 15 cali, 300/250 Zn/s)	1350	40
HOP-40 (24 igły, 10 cali, 200 Zn/s)	990	40
HQP-45 (24 igły, 15 cali, 200 Zn/s)	1450	40
Drukarka Laserowa (6 str/min.)	3990	80
Kasety do drukarek 10" (120 N)	134.5	20
10" (przy 10 szt.)	130	20
15" (przy 10 szt.)	180	20
Kable podłączeniowe do komputerów	20	

B) PLOTTERY SECONIC

SPL 410 (DIN A3)	1650	40
SPL 430 (DIN A3)	1650	40
SPL 450 (DIN A3)	2050	40
SPL 600 (DIN A2)	5900	80
SPL 800 (DIN A1)	6900	120
SPL 1000 (DIN A0)	12500	160

C) MONITOR GRAFICZNY CONRAC 19" 5900 120

D) TERMINALE

ESPRiT OPUS 220	849	40
ESPRiT OPUS 2	649	40
ESPRiT OPUS 4	849	40

E) STREAMERY ARCHIVE

Archive FT 60 + cartridge	1310	40
ArchiveST600 + 1 cartridge	1310	40

F) DYSKI SEAGATE

Seagate 20 MB ST 225	399	40
Seagate 40 MB ST 251	639	40
Seagate 20 MB ST 225 z kontr, i kablem	497	40
Seagate 40 MB ST 251 z kontr, i kablem	749	40

G) DYSKIETKI MAXELL

51/4 cala		
MD1-D przy 1000 szt.	1,23	150
MD2-D przy 1000 szt.	1,58	150
MD1-DD przy 1000 szt.	1,94	150
MD2-DD przy 1000 szt.	1,99	150
MD2-HD przy 1000 szt.	3,09	150
3,5 cala		
MF1-DD przy 1000 szt.	2,46	150
MF2-DD przy 1000 szt.	2,52	150
MF2-HD przy 1000 szt.	6,15	150
3 cale		
CF2 przy 1000 szt.	4,45	150

WARUNKI HANDLOWE:

- Oferowany sprzęt zamówić można korespondencyjnie dokonując przelewu z konta A telegraficznie na nasze konto bankowe.
- Do sumy każdego przelewu prosimy dodać DM 10,- na pokrycie kosztów przelewu bankowego.
- Po dokonaniu przelewu prosimy o wysłanie do nas kopii dowodu wpłaty wraz z dokładną specyfikacją.
- Ceny należy rozumieć jako ceny z naszego składu w Munchen (FO B Munchen).
- Przy większych zakupach udzielamy rabatów.

NINIEJSZYM ZAMAWIAM:

1. .... DM ....
2. .... DM ....
3. .... DM ....

Koszty manipulacji bankowych .... **DM 10,-**

Razem **DM** ....

Załączam czek lub kopię zlecenia bankowego na przelew w/w sumy na konto nr 7137320 w Beyerische Vereignsbank Munchen BLZ 70020270 zrealizowanego w dniu. ....../....../....... przez bank ...., oddział............... w ....
Podpis wpłacającego................ Nazwisko i imie .... Data ....

NAZWISKO I IMIĘ ODBIORCY..
ADRES ODBIORCY ....

SPRZĘŻENIE ZWROTNE

Drogi Bajtku!

Na listy Czytelników odpowiada Marcin Waligórski

Od początku roku głośno jest o nowym języku Turbo Pascal 4.0. Czy jest to faktycznie język tak dobry, jak wskazują na to pierwsze publikacje m.in. w naszej prasie? (Komputer nr 6/88)?' Na czym dokładnie polega niezgodność z Turbo Pascalem w wersji 3? Co dokładnie zyskuje programista?

Czy Turbo Pascal 4.0 został wydany w wersji dla komputerów ośmiobitowych — konkretnie Amstrada PCW 8512?

Od początku roku głośno jest o nowym języku Turbo Pascal 4.0. Czy jest to faktycznie język tak dobry, jak wskazują na to pierwsze publikacje m.in. w naszej prasie? (Komputer nr 6/88)?' Na czym dokładnie polega niezgodność z Turbo Pascalem w wersji 3? Co dokładnie zyskuje programista?

Czy Turbo Pascal 4.0 został wydany w wersji dla komputerów oś- miobitowych — konkretnie Amstrada PCW 8512?

Wśród programistów zawodowych z którymi mam styczność Turbo Pascal 4.0 wyparł starszą wersję 3.01A niemal błyskawicznie. Co więcej, nikt ze znanych mi użytkowników tego kompilatora nie wyrażał jak dotąd chęci powrotu do pracy ze starym Turbo Pascal-em, co uważam za znaczący syndrom jakości nowego programu. Faktycznie, porównanie wychodzi zdecydowanie na jego korzyść; jedyna rzecz, co do której można mieć wątpliwości, to prostota obsługi — mnogość opcji do wyboru z rozwijanych menu i wieloczęściowa organizacja ekranu sprawiają, że osobie niewprawnej trudno byłoby po prostu usiąść i zacząć pracę z programem. Konieczne jest wcześniejsze przestudiowanie podręcznika.

Sam język Turbo Pascala odbiega bardzo znacznie od prototypu opracowanego przez Niklausa Wirtha. Można stwierdzić, że standardowy Pascal posłużył tu jedynie za szkielet języka, który obudowano mnóstwem standardowych procedur, nowych rodzajów deklaracji i definicji. Trzeba powiedzieć, że odejście, lub jak kto woli — rozwinięcie standardu Pascala przeprowadzono zdecydowanie. W Turbo Pascal-u do wersji 3.0 włącznie dało się bez większych problemów skompilować i uruchomić program napisany w standardzie Pascal-a; Turbo 4.0 takiej rzeczy zrobić nie pozwoli. Nie da się również przenieść programów napisanych w Turbo 3.0. Producent do nowego kompilatora dołączył jednak program użytkowy — swoisty translator — który częściowo tłumaczy programy na dialekt 4.0, częściowo zaś informuje o możliwości wystąpienia w programie błędu. Na przykład częstym źródłem trudności przy translacji są deklaracje inline, w których odwołujemy się do systemu wejścia-wyjścia; w Turbo 4.0 został on całkowicie zmieniony.

W programie źródłowym od razu rzuca się w oczy deklaracja uses, w której podaje się linkerowi nazwy modułów, które mają być z programem połączone. Przeniesienie zleceń linkera do programu źródłowego jest oryginalnym pomysłem Borlanda, i wydaje się, że nie jest on pozbawiony słuszności. Poza tym istnieje cała masa funkcji standardowych języka, w tym funkcje dające dostęp do DOS-a i procedury obsługi grafiki. Wprowadzono też oprócz rozszerzeń zmiany. Nie da się już użyć read (Kbd, Ch) — plik Kbd znikł, zamiast niego są nowe funkcje odczytu stanu klawiatury. Pełny opis różnic wykracza poza łamy tej rubryki, ale wróćmy do tego tematu na łamach „Bajtka".

Produkowany przez nowy kompilator kod wynikowy jest szybszy niż w przypadku starszych wersji języka. Przede wszystkim zaś produkowane są programy typu EXE, co pozwala na przekroczenie rozmiaru 64KB, który tak ograniczał profesjonalistów w przypadku Turbo 3.0. Nowy zintegrowany linker pozwala na konsolidację kilku skomplikowanych plików, nawet gdy były kompilowane np. przy użyciu Turbo C.

Kod wynikowy jest budowany poprawnie — nie spotkałem się jeszcze z przypadkiem wystąpienia błędu w generacji kodu. Jest to zresztą charakterystyczne chyba dla wszystkich kompilatorów marki Borland.

Reasumując, „przesiadka na Turbo Pascal 4 0 wydaje się być grą wartą świeczki. Smutną wiadomością może być to, iż według wszelkich znaków na niebie i ziemi posiadacze komputerów 8-bitowych zostaną tej przyjemności pozbawieni. Sam kompilator wymaga więcej pamięci przy pracy niż wynosi TPA w jakimkolwiek systemie CP/M. Poza tym największy atut Turbo 4.0, którym jest przekroczenie bariery 64KB kodu. nie ma żadnej przyszłości w przypadku systemu 8-bitowego..

Czy Atari 1040ST to to samo, co 512ST + 0,5 MB dodatkowej pamięci? Czy w związku z tym (i jak) możliwa jest przeróbka 520ST na 1040ST?

Krzysztof Miklas
ul. Bryły 3/76
Warszawa

Nie jest to całkiem to samo. 1040ST ma zmienioną budowę wewnętrzną, zaś na płycie 520ST brak miejsca na rozszerzenie pamięci. Rozszerzenie jest jednak możliwe (i opłaca się później np. większą wygodą pracy z dużym RAM-dyskiem); najlepiej przeróbki dokonać przez wymianę układów pamięci — choć jest to inwestycją dość kosztowną.

Piszę do „Bajtka", aby poruszyć sprawę jakby dotąd przez nikogo nie dostrzeganą. Chodzi o język i styl, części wydawanych u nas książek z dziedziny informatyki. Udostępnianie wiedzy w tej dziedzienie, jak w wielu innych jest wielką sprawą i w związku z tym komunikatywność powinna być przedmiotem specjalnej troski. W zasadzie jest tak w przypadku czasopism tej dziedzinie poświęconym, z których za najlepsze uważam „Mikroklan", „Bajtek" i „Komputer". Czytelnik tych pism wcześniej czy później będzie starał się jednak sięgnąć do książek — a tu sytuacja jest niekiedy wręcz fatalna!

Wiem, że trudności wydawnicze limitują ilość dostępnych na rynku tytułów. To jednak nie tłumaczy dlaczego te, które się ukazują, i to w masowym nakładzie, nie są pisane w sposób jasny dla przeciętnego czytelnika? Ukazało się sporo książek popularyzatorskich — np. D. Madej, K. Marasek, K. Kuryłowicz „Komputery osobiste"; tej samej spółki „Przewodnik po ZX Spectrum"; Rolanda Wacławka „Mój komputer ZX Spectrum" — ale zawierają one jedynie podstawową wiedzę.

Książki pisane przez profesjonalistów są zaś w stopniu zniechęcającym trudne w odbiorze. I to bynajmniej nie z powodu treści, lecz z przyczyny napuszonego stylu, przenaukowienia, usystematyzowania służącego wszystkiemu innemu, tylko nie zrozumieniu tekstu przewagi teorii nad przykładami, itp. itd. Innymi słowy, takie książki, jak: J. Bielecki „Język C — interpretacja standardu", tegoż autora „Oprogramowanie mikrokomputerów" i „Turbo Pascal wersja 3.0", S. Gasik, P. Kulczycki, K. Pisaecki, J. Witaszek „PL/I/F/, M. Iglewski, J. Madey, S. Matwin „Pascal", sprawiają wrażenie pisanych przez naukowców dla naukowców. Ale skoro tak, to powinny być wydawane w mniejszym nakładzie w postaci skryptów!

Czytałem sporo publikacji w języku angielskim — b. trudno dostępnych, ale cóż? — i prawie w 100 proc. uważam je za bardziej komunikatywne od ich polskich odpowiedników. W USA jak widać od autorów wymaga się, aby ich prace były zrozumiałe dla jak największej grupy odbiorców, bo to wiąże się z ilością sprzedanych książek. Jak jest u nas — wiadomo, ale to niczego nie tłumaczy.

Ma Pan rację, z wyjątkiem jednego: publikacje przeznaczone jako lektura dla naukowców powinny być w tym samym stopniu czytelne i zrozumiałe, jak wszystkie inne. Żeby się przekonać, jak powinna wyglądać wzorowa książka o przedmiocie ścisłym (do takich można zaliczyć większość publikacji z dziedziny informatyki), nie trzeba znać angielskiego. Wystarczy wziąć do ręki wydane w serii BIO tłumaczenia: I.O. Angel „Wprowadzenie do grafiki komputerowej"; Dennie van Tassel „Praktyka programowania" (bardzo stara książka, warto zobaczyć, jak jest napisana) czy słynną pracę Kernighana i Richiego „Język C". Zwłaszcza ta ostatnia pozycja zasługuje na uwagę. Książka, która jest przyjęta jako oficjalny raport standardu języka C, może być bez trudu zrozumiana przez licealistę!

Widocznie w naszych publikacjach wręcz odwrotne cechy wynikają chyba z faktu, iż autorzy o pewnym dorobku naukowym boją się narazić swojej reputacji poprzez publikację książki, nie mającej cech „naukowości". Czy jest to wywołane rzeczywistą presją środowiska naukowego — nie wiem. Zjawisko nie jest natomiast nowe; od długiego czasu walczy z nim np. red. Roland Wacławek. Wspomniał on o tym również w wywiadzie opublikowanym na łamach „Bajtka".

Osobny rozdział stanowią publikacje autorstwa Jana Bieleckiego. Autor posługuje się w nich własną, odrębną od przyjętej powszechnie, terminologią, co oczywiście nie wpłynęło dodatnio na zrozumiałość jego tekstów. I jakkolwiek pan Bielecki wyjaśnił na łamach „Mikroklanu" zasady, rządzące jego wersją terminologii, to jednak nie są one przekonywujące. Łatwo pokazać, że przystają one jedynie do x tzw. klasycznych języków programowania, co w przyszłości może być źródłem dodatkowego zamętu. Poza tym wiele elementów rozumowania autora „Oprogramowania mikrokomputerów" jest niejasnych. Dlaczego np. zmienna ma być obiektem istniejącym jedynie w programie wynikowym? Przecież program źródłowy, zapisany jedynie w innym języku programowania? Czym jest w takim razie zmienna w przypadku języka interpretowanego? Podobne pytania można by mnożyć.

Sumując, informatyka nie jest od dłuższego czasu „ziemią niczyją" i całkowite zrywanie z kanonami zdaje się przynosić więcej szkody, niż pożytku. Poza tym o ile pisanie samo w sobie jest rzemiosłem, to dobre pisanie — sztuką. Nie każdy autor jest artystą, ale w kompetencji wydawnictw leży wpływ na postać końcową wydawanych książek.

Komputer Amstrad 464 posiada tylko jedno gniazdo manipulatora. Firma przewiduje możliwość podłączenia dwóch drążków sterowych, ale w nietypowy sposób — drugi jest dołączony do gniazda w podstawie drążka pierwszego. Drążki normalnie dostępne w handlu nie posiadają dodatkowego gniazda. Jak zatem wyglądałby schemat podłączenia dwóch manipulatorów do tego komputera?

Jan Boruc
Poznań

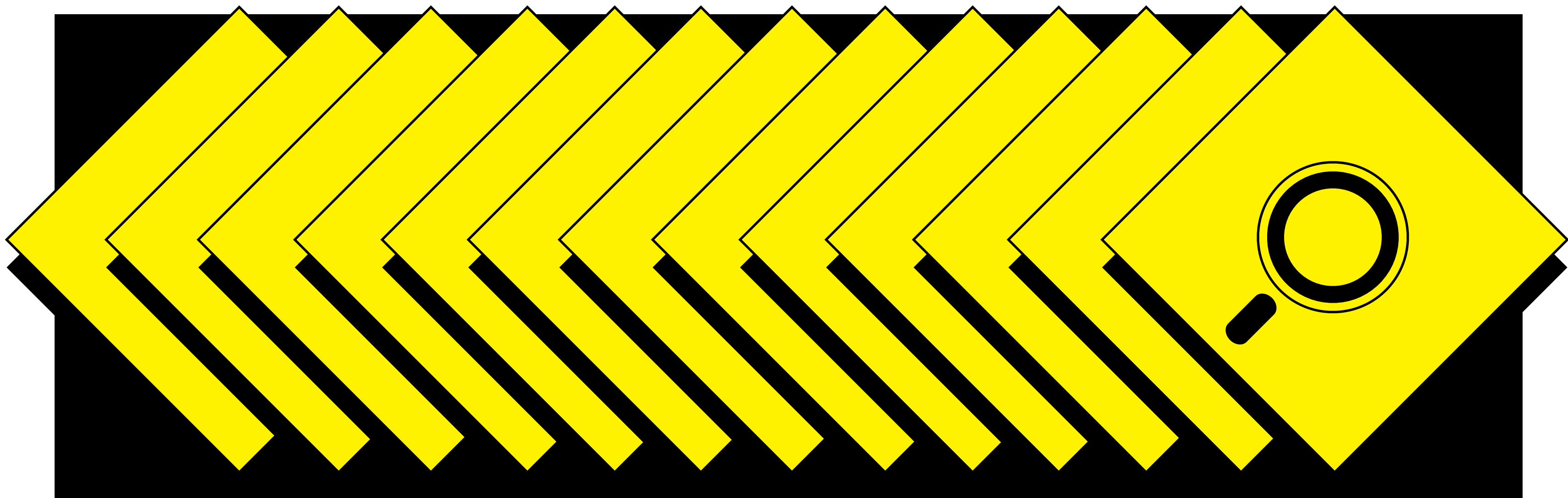
Schemat podłączenia dwóch drążków sterowych opublikowaliśmy już w „Bajtku" nr 5-6/1986. Przeróbka polega jedynie na właściwym podłączeniu przewodów, zatem jest dostępna dla każdego.

Przy uruchamianiu programów w Basicu ZX Spectrum zwłaszcza tych większych, przydałby się program, pozwalający na wyszukanie wszystkich wystąpień danej zmiennej w programie. Funkcja ta jet niestety niedostępna w Basicu. Czy można prosić o informację, w jaki sposób taką procedurę zrealizować w języku maszynowym?

(nazwisko i adres do wiadomości redakcji)

Nie jest konieczne używanie kodu maszynowego. Program da się napisać w Basicu, w oparciu jedynie o znajomość struktury zapisu programu w pamięci. Rozwiązanie takie znaleźć można w programie „Edytor 9888". Bajtek 12/1987 s. 9. Program ten realizuje również inne funkcje pomocne przy edycji i uruchamianiu programu.

Marcin Waligórski



KONKURS
O ZŁOTĄ DYSKIETKĘ "BAJTKA"

Zgodnie z zapowiedzią ogłaszamy dzisiaj drugą edycję konkursu klubów mikrokomputerowych „O Złotą Dyskietkę BAJTKA".

Kluby komputerowe powstają już nie tylko w dużych miastach. Ich formy organizacyjne bywają bardzo różne. Wydaje się wielu osobom, że wystarczy postawić w świetlicy komputer — a leszcze lepiej kilka — i mamy już znakomity klub komputerowy. Niestety, nie jest to takie proste. Uzyskanie sprzętu to dopiero początek...

Działacze klubów komputerowych formułują wiele postulatów mających pomóc w ich pracy. Dotyczą one dostępności literatury, przygotowywania instruktorów do pracy z młodzieżą, serwisu sprzętu, form wymiany doświadczeń, itd.

Chcąc pomóc tym wszystkim, którym zależy na rozwoju ruchu klubów mikrokomputerowych ogłaszamy nasz konkurs. Jego celem jest stworzenie możliwości szerokiej wymiany doświadczeń i informacji, popularyzacja najciekawszych pomysłów dotyczących organizacji pracy w klubie i jego działalności społecznej. Organizatorzy klubów stają przed podobnymi problemami: jak zorganizować zajęcia, czego uczyć, pokazywać gry komputerowe, czy też udawać, że takie nie istnieją, skąd brać fundusze na potrzeby klubu, ... i tak w kółko.

A przecież inni zdołali już wcześniej odpowiedzieć na te pytania i — co ważniejsze — przetestowali przyjęte rozwiązania w praktyce. Po co więc wyważać otwarte drzwi? Chcemy pokazać najciekawsze formy działalności i tym samym dać bogatszą możliwość wyboru kierunku pracy nowym klubom a także wzbogacić formy działania klubów o dłuższym stażu

Do konkursu zapraszamy wszystkie kluby mikrokomputerowe: szkolne, zakładowe, działające pod patronatem organizacji młodzieżowych, domów kultury itd. działające w Polsce, a także — to nowość — przy polskich placówkach za granicą. Zgłoszenia mogą przysyłać także kluby nieformalne, których działalność opiera się na zasadach towarzyskich.

W rywalizacji będą brane pod uwagę następujące kryteria:

— społeczne oddziaływanie klubu
— dostępność i otwarty charakter klubu
— wymierne efekty działania (np. samodzielne tworzenie oprogramowania)
— działalność edukacyjna i popularyzatorska w środowisku
— samodzielność w zdobywaniu sprzętu i materiałów pomocniczych
— prowadzenie szkolenia na różnych poziomach wiedzy
— nowatorskie formy działania

Ażeby wziąć udział w konkursie należy przesłać do naszej redakcji:

— pisemne zgłoszenie udziału w konkursie wg. wzoru zamieszczonego poniżej
— krótkie, syntetyczne opisanie dotychczasowej działalności (maksymalnie 1 stiona maszynopisu)
— plan zamierzeń na rok szkolny 1988/89 (również do 1 strony maszynopisu)

Na zgłoszenia do konkursu czekamy do 30 listopada 1988 r.

W ciągu kilku miesięcy komis|a konkursowa, w skład której wejdą członkowie redakcji i przedstawiciele instytucji sponsorujących, odwiedzi kluby i wybierze najlepsze z nich. Będą one prezentowane na łamach „Bajtka" „Sztandaru Młodych".

Wielki Finał, w którym współzawodniczyć będą najlepsi z najlepszych odbędzie się w połowie przyszłego roku i będzie transmitowany przez telewizję. Zwycięzca otrzyma pamiątkową „Złotą Dyskietkę BAJTKA" oraz komputer ufundowany przez Przewodniczącego Komitetu ds. Młodzieży i Kultury Fizycznej.

Współorganizatorami konkursu i fundatorami nagród są również:

Urząd Postępu Naukowo-Technicznego i Wdrożeń
Centralne Biuro TMMT ZSMP
Związek Młodzieży Wiejskiej
Zrzeszenie Studentów Polskich

Czekamy na zgłoszenia!

Nasz adres: „Bajtek" ul. Wspólna 61 00—687 Warszawa, Konkurs „O Złotą Dyskietkę BAJTKA"

Przypominamy

Główną nagrodę w ubiegłorocznej, pierwszej edycji konkursu O ZŁOTĄ DYSKIETKĘ „BAJTKA" zdobył klub MERIZAP z Ostrowa Wlkp. działający przy Zespole Szkół Technicznych MERIZAP. Nagrodę dla MERIZAP-u, komputer Amstrad 6128 ufundował Minister ds. Młodzieży Aleksander Kwaśniewski.

Drugie miejsce i komputer Timex ufundowany przez Turniej Młodych Mistrzów Techniki ZSMP otrzymał Kiub Mikrokomputerowy przy Pałacu Młodzieży w Szczecinie.

Na trzeciej pozycji znalazł się klub FENIKS, działający przy Miejsko-Gminnym Ośrodku Kultury w Reczu. Nagrodę dla niego ufundował Związek Młodzieży Wiejskiej.

Nagrodę czwartą fundowaną przez Centralną Składnicę Harcerską zdobył klub Harcbajt działający przy komendzie Horągwi ZHP w Gdańsku

Przyznanych zostało także pięć wyróżnień dla klubów:

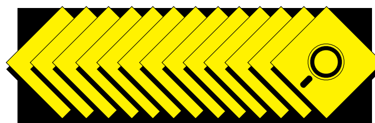
SYNTAX ERROR z Koszalina — półprywatny klub działający pod patronatem ZMW,

MANIAK z Warszawy — działający przy Klubie Osiedlowym na Ursynowie,

BIT z Warszawy — jedyny w swoim rodzaju klub rodzinny,

Złoty Amstrad ze Złotego Stoku — działający pod patronatem Zakładów Tworzyw i Farb Złoty Stok,

Mr ATARI z Kozienic — całkowicie prywatny klub zrzeszający ośmiu stałych i dziesięciu korespondencyjnych członków.



ZGŁOSZENIE DO KONKURSU ZŁOTĄ DYSKIETKĘ "BAJTKA"

Nazwa klubu: . . . .

Opiekun (szkoła, zakład pracy, itp.): . . . .

. . . .

Adres klubu: ul. . . . .

kod . . . . miejscowość: . . . .

Numer telefonu (klubu, prezesa, ew. jednego z członków): . . . . .

Nasz klub działa od . . . . roku, obecnie zrzesza . . . .

. . . . członków

Zgłaszamy nasz klub do współzawodnictwa O ZŁOTĄ DYSKIETKĘ „BAJTKA".

W załączeniu przesyłamy:

– Opis działania naszego klubu.

– Zamierzenia na rok 1988/89

Prezes klubu: . . . .

Adres (telefon): . . . .

. . . . podpis . . . .

Opiekun (podpis i pieczątka) . . . .

ZABAWA W SKLEP



Cześć Maluchy!

Proponuję Wam dzisiaj przerobienie komputera na kasę sklepową. Będzie to jednak nieco mądrzejsza kasa niż te, które widujemy na codzień.

Zwykła kasa — pomimo bardzo poważnego wyglądu — potrafi bardzo niewiele. Praktycznie jedynie dodaje wystukiwane na klawiaturze sumy pieniędzy. Pomocnicze obliczenia (na przykład ile kosztuje 62 dekagramy sera w cenie 347 złotych za kilogram) musi dokonywać sprzedający. Spróbujmy mu pomóc.

Człowiekowi obliczenie rachunku w rodzaju
0.62*347
zajmuje przeciętnie kilkanaście sekund, dla komputera natomiast jest to czas niezauważalny. Trzeba mu jednak te dane wpisać. Najprostszym sposobem byłoby każdorazowe informowanie komputera o ilości i cenie jednostkowej (na przykład za kilogram, litr lub sztukę) towaru. Wówczas następowałoby obliczenie należności i zsumowanie jej z poprzednimi. Zastanówmy się jednak, czy nie można jeszcze bardziej ułatwić życia sprzedawcy.

Ilość towaru (w odpowiednich jednostkach) będzie trzeba podawać za każdym razem, bo zależy ona jedynie od życzenia klienta. Bez trudu możemy natomiast przewidzieć jednostkowe ceny towarów, i to może za nas pamiętać komputer. W takim przypadku wystarczy podać nazwę sprzedawanego towaru i jego ilość. Dla uniknięcia nieporozumień komputer powinien pamiętać, że ilość mleka mierzy się w litrach, cukru w kilogramach a bułek w sztukach.

Wiedząc jak ma działać nasz program możemy przystąpić do jego konstruowania. Najpierw zdeklarujemy wielkość tablic (linia 10). Przyjęliśmy, że w naszym sklepie będzie maksymalnie 50 różnych towarów. Oczywiście możecie tę liczbę zmienić dowolnie, na ile pozwoli Wam pamięć komputera. W tablicy **t$(50)** przechowywać będziemy nazwy towarów, w tablicy **j(50)** liczby oznaczające w jakich jednostkach mierzony jest dany towar. Kolejne liczby oznaczać będą następujące jednostki:

1 — kilogramy
2 — litry
3 — sztuki
4 — opakowania

Nazwy jednostek przechowywane są w tablicy **n$(4,1)**. Posiada ona dwa wymiary, gdyż w rozmowie ze sprzedawcą komputer posługuje się nazwami jednostek w dwóch przypadkach gramatycznych.

Ceny przechowywane są w tablicy **t(50)**.

W liniach 20—70 następuje zapisanie tablicy **n$(4,1)** danymi z linii 80. Następnie komputer przechodzi do odczytywania informacji o towarach zapisanych w liniach 1000—2000. Zmienna n, przyjmująca wartość początkową **1** (linia 100) oznacza liczbę towarów, które komputer pamięta. Zmienna **s**, przyjmująca wartość początkową **0** (linia 110) oznacza sumę należności za sprzedane towary.

W pierwszej kolejności odczytywana jest nazwa towaru I umieszczana w odpowiednim miejscu w tablicy **t$(50)** (linie 120 i 130). Za każdym razem sprawdzane jest, czy wczytane słowo nie jest słowem **"koniec"** (linia 140), a jeśli tak, to komputer wyskakuje z pętli do linii 200 i przechodzi do wykonywania właściwego programu. Każde inne słowo przyjmowane jest jako nazwa towaru i następuje wczytanie liczb oznaczających jednostkę miary i cenę towaru (linie 150—170). Zmienna n zwiększana jest o 1 i następuje powrót do początku pętli wczytywania czyli linii 120 (linie 180—190).

Informacje o towarze zapisane są w liniach 1000- -2000. Program przystosowany jest do tego, aby można było w prosty sposób uzupełniać listę towarów. Dlatego właśnie wczytywanie nie odbywało się przy pomocy pętli **FOR... NEXT** (w tym przypadku musielibyśmy określić liczbę towarów). Sygnałem, że należy zakończyć wczytywanie jest słowo **"koniec"** i jest to jedyne słowo, którego w liniach danych nie można zmienić.

Po wyskoczeniu z pętli program przechodzi do wykonania linii 195, w której następuje zmniejszenie wartości **n** o jeden, tak, aby słowo "**koniec**" nie zostało policzone jako nazwa towaru. Teraz następuje zasadnicza część programu. W linii 200 czyszczony jest ekran, a w liniach 300 i 310 wczytywana nazwa towaru (komputer oczekuje na wprowadzenie nazwy z klawiatury). Po potwierdzeniu klawiszem **ENTER/RETURN** komputer sprawdza, czy wprowadzono jakąś nazwę (linia 320), w przeciwnym przypadku — jeśli wprowadzono zbiór pusty "" — przechodzi do wydrukowania sumy końcowej.

Dowolny, różny od pustego ciąg znaków przyjmowany jest jako nazwa towaru i w pętli (linie 330—350) porównywany z zawartością tablicy nazw **t$(50)**. Jeśli komputer natrafi na identyczną nazwę, wyskakuje z wnętrza pętli do podprogramu obliczającego należność zachowując numer towaru pod zmienną i.

Może się zdarzyć, że wpisana nazwa towaru nie zostanie znaleziona w pamięci. W takim przypadku komputer po zakończeniu pętli przeszukiwania przejdzie do podprogramu wprowadzania informacji o nowym towarze. Najpierw komputer zwiększa zmienną **n** o jeden (przybywa bowiem nowy towar) prosi o informację, jakiej jednostki używamy do mierzenia tego towaru (linie 400—490). Następnie pyta o cenę za jedną jednostkę miary. Pytając korzysta z zawartości tablicy **n$(4,1)**.

Odpowiednie dane umieszczone zostają w tablicach pod numerem **n**. Odtąd komputer będzie dysponował jeszcze tę dodatkową informacją.

Podprogram obliczania należności wywoływany jest zarówno w przypadku odnalezienia nazwy to waru jak i w przypadku uzupełnienia danych. Rozpoczyna się od pytania o ilość towaru (linie 600 610). Również w tym przypadku komputer korzysta z nazw zawartych w tablicy **n${4,1)**. W następnej kolejności obliczana jest należność za zakupioną ilość towaru (zmienna **c**, linia 620) i suma do zapłaty za wszystkie dotychczas wpisane towary (zmienna **s** w linii 630). Na ekranie ukazuje się jedynie wartość zmiennej **c** (linia 640). Po instrukcji **PRINT** występują dwa przecinki, co powoduje, że wartość zmiennej **c** drukowana jest bliżej prawej strony ekranu

Na tym kończy się główna pętla programu. Z tego miejsca następuje skok do linii 300 i wczytywanie nazwy kolejnego towaru.

Ostatni podprogram, to drukowanie sumy końcowej. Wywoływany jest on wówczas, gdy zamiast podania nazwy kolejnego towaru naciskamy po prostu **ENTER/RETURN** (patrz linia 320). Po wydrukowaniu sumy komputer natrafia na instrukcję **INPUT r$** (linia 710) a więc zatrzymuje się do chwili wciśnięcia klawisza **ENTER/RETURN**. Następnie zeruje zmienną s (linia 720) i skacze na początek programu do linii 200 (linia 730), co oznacza, że możemy przystąpić do obliczenia ceny kolejnej partii towaru.

* * *

Oczywiście, to co dzisiaj zrobiliśmy nie jest naszym odkryciem. Kasy sklepowe działające na podobnych zasadach można zobaczyć już w Polsce. Te prawdziwe kasy są jeszcze mądrzejsze. Mogą być na przykład połączone z wagą i wówczas sprzedawca musi jedynie wprowadzić informację jaki towar położył na wagę, a reszta dzieje się automatycznie.

Niektóre towary kasa może rozpoznawać samodzielnie. Dzieje się to za sprawą tak zwanego kodu prążkowego. Wystarczy przesunąć specjalnym czytnikiem w poprzek tych prążków i symbol towaru jest odczytany. Sprawdzenie aktualnej ceny i podsumowanie odbywa się podobnie jak w naszym programie.

To jeszcze nie wszystko. Kasy potrafią się komunikować między sobą, jak również z centralnym magazynem, w którym towar składowany i wydawany jest całkowicie automatycznie. Tak więc w każdej chwili centralny komputer posiada informacje o aktualnym zapasie towarów. Jeśli zapas któregoś z towarów zmniejsza się, komputer sam zamawia dalszą partię u producenta. Jeśli zaś producentem jest całkowicie zrobotyzowana fabryka, komputer w sklepie może wydać polecenie zmiany profilu produkcji na bardziej w danej chwili potrzebną.

To nie jest fragment powieści fantastyczno-naukowej, to nawet nie są przewidywania niedalekiej przyszłości, to dzieje się już dzisiaj!

O czym zapewnia Was

Romek

```
9 REM ******* dane poczatkowe *******
10 DIM t$(50),j(50),c(50),n$(4,1)
20  FOR i=1 TO 4
30    FOR j=0 TO 1
40     READ r$
50     LET n$(i,j)=r$
60    NEXT j
70  NEXT i
80 DATA "kilogram","kilogramow","litr"
,"litrow","sztuke","sztuk","opakowanie
","opakowan"
99 REM **** informacja o towarze ****
100 LET n=1
110 LET s=0
120  READ r$
130  LET t$(n)=r$
140  IF t$(n)="koniec" THEN GOTO 195
150 READ t1,t2
160 LET j(n)=t1
170 LET c(n)=t2
180 LET n=n+1
190 GOTO 120
195 LET n=n-1
199 REM ******* petla glowna *******
200 CLS
299 REM **** poszukiwanie towaru ****
300 PRINT "Nazwa towaru"
310 INPUT t$
320 IF t$="" THEN GOTO 700
330  FOR i=1 TO n
340   IF t$=t$(i) THEN GOTO 600
350 NEXT i
399 REM ******** nowy towar *********
400 PRINT "Brak danych o tym towarze."
410 PRINT "Podaj jednostk miary:"
420 PRINT "1 - kilogram"
430 PRINT "2 - litr"
440 PRINT "3 - sztuka"
450 PRINT "4 - opakowanie"
460 LET n=n+1
470 LET t$(n)=t$
480 INPUT r
490 LET j(n)=r
500 PRINT "Podaj cene za 1
";n$(j(n),0)
510 INPUT r
520 LET c(n)=r
530 LET i=n
599 REM *** obliczenie naleznosci ***
600 PRINT "Ile ";n$(j(i),1)
610 INPUT m
620 LET c=m*c(i)
630 LET s=s+c
640 PRINT ,,c
650 GOTO 300
699 REM ******* suma koncowa *******
700 PRINT "Razem",,s
710 INPUT r$
720 LET s=0
730 GOTO 200
999 REM ****** dane o towarze ******
1000 DATA "cukier", 1, 165, "jajka",
3, 32, "mleko", 2, 34
1010 DATA "maslo", 4, 225, "makaron",
4, 123, "jablka", 1, 310
1020 DATA "buka", 3, 12, "chleb", 3,
36 , "herbata", 4, 153
1030 DATA "lody", 3, 43, "cukierki",
1, 315, "sok", 2, 162
1040 DATA "czekolada", 4, 455,
"kompot", 2, 335, "budy", 4, 32
2000 DATA "koniec"
```




Wypróbowywanie marsochoda w najtrudniejszych warunkach terenowych

dokończenie ze str. 32

Wiadomo już dzisiaj, że radziecki aparat wystartuje z Bajkonuru we wrześniu 1992 roku i po około 350 dniach dotrze w okolice Marsa, najpierw „Kolumb" wejdzie na orbitę wokółmarsjańską, a następnie z jego pokładu rozpocznie się desant ładownika.

Przebieg lądowania będzie następujący: na wysokości 13 kilometrów otworzy się spadochron. Na wysokości 9 km zrzucone zostaną na powierzchnię Marsa penetratory (2 sztuki). Zagłębiwszy się w grunt, będą działały na Marsie od roku do czterech lat. Kilka penetratorów (najprawdopodobniej polecą w kierunku Marsa dwa Kolumb utworzy sieć umożliwiającą długotrwałe pomiary sejsmiczne.

Po zrzuceniu penetratorów, z ładownika wypuszczony zostanie aerostat. Zdjęcia telewizyjne wykonywane przez umieszczoną w aerostacie kamerę będą miały rozdzielczość kilku centymetrów. W gondoli aerostatu znajdzie się także zestaw meteorologiczny do pomiaru temperatury, ciśnienia, wilgotności, prędkości wiatru itp. Aerostat będzie się poruszał jedynie w ciągu marsjańskiego dnia, robiąc po 100—400 km. W nocy. będzie opuszczał się na powierzchnie. W ostatniej fazie zbliżania się ładownika do powierzchni planety siła uderzenia zostanie zamortyzowana poprzez krótkie włączenie specjalnych silników hamujących (spadochron zostanie wcześniej odstrzelony).

Po ustabilizowaniu się ładownika na powierzchni i wykonaniu niezbędnych obserwacji — z jego pokładu zjedzie na powierzchnię planety marsochód. Zasiąg ruchu marsocnodu wyniesie około 300 km. Jego prędkość zależeć będzie od zasilania w energię oraz od rzeźby terenu marsjańskiego i programu naukowego badań. Źródłem energii marsochodu będą baterie słoneczne lub izotopy.

Marsochód zostanie wyposażony w zestaw przyrządów do badania gruntu marsjańskiego, a także do badań sejsmicznych, pomiaru pól magnetycznych i elektrycznych, oceny aktywności biologicznej próbek. Aby ten unikalny pojazd mógł właściwie wypełnić swoje zadanie potrzebne są niezliczone próby prototypów. Przeprowadzane są one, w różnych regionach Związku Radzieckiego. Chodzi o to, aby marsochoda sprawdzony został w warunkach maksymalnie zbliżonych do marsjańskich.

Uczeni wiele sobie obiecują po „Kolumbie". Podobnie jak wielki poprzednik, który w 1492 roku dopłynął do Ameryki, również jego mechaniczny następca — wyruszający w swą podróż dokładnie 500 lat po tamtym historycznym wydarzeniu — odkryje przed ludzkością nowe horyzonty. Wiele faktów w każdym razie na to wskazuje!

Co dalej po „Kolumbie" ? Kolejny skok polegał będzie na dostarczeniu na Ziemię próbek marsjańskiego gruntu. Następny etat to wysłanie w latach 2000—2005 ciężkich marsochodów o długim okresie działania i większym zasięgu. Dalej, około roku 2010, odbędzie się kombinowana wyprawa z wysadzeniem marsochodów nowego typu i pobraniem dużych partii gruntu z różnych miejsc, odległych o setki kilometrów.

I wreszcie, około roku 2020, powstaną warunki do zorganizowania pilotowanej wyprawy ludzi na Marsa, z wyjściem kosmonautów na powierzchnię. Oczywiście, gdyby się udało doprowadzić do ściślejszej współpracy międzynarodowej w badaniu kosmosu, przede wszystkim radziecko-amerykańskiej, to wszystkie podane wyżej terminy mogłyby ulec daleko idącemu skróceniu!

Jurij Zajcew

Schemat ostatniej fazy lotu misji „Kolumb"

PROJEKT KOLUMB

Wreszcie się zaczęło! 7 i 12 lipca z kosmodromu Bajkonur wystartowały w kierunku Marsa dwie radzieckie sondy kosmiczne „Fobos-1" i „Fobos-2". W połowie przyszłego roku przelecą one w odległości kilkudziesięciu metrów (!) nad powierzchnią Fobosa, tajemniczego księżyca Marsa, badając różnymi metodami jego powierzchnię.

Wyprawa „Fobosów" jest wstępem do porywającej wyobraźnię ekspedycji Człowieka na Czerwoną Planetą. Zanim jednak do tego dojdzie — ostrożni uczeni wymieniają rok 2020 jako w pełni realną datę lądowania Homo Sapiens na Marsie — konieczne są „kolejne przybliżenia" służące zdobyciu doświadczeń, wypróbowaniu sprzętu itp. Jednym z takich kluczowych „pośrednich" przedsięwzięć jest radziecki projekt „Kolumb". Pisze o nim specjalnie dla Czytelników „Bajtka" nasz współpracownik z Instytutu Badań Kosmicznych w Moskwie — Jurij Zajcew.

Wiemy już o Marsie sporo. Dane przekazane przez radzieckie i amerykańskie sondy mars- jańskie są całkowicie wiaragodne. Właśnie dzięki nim mogliśmy zdecydowanie odrzucić tezę o ewentualnym sztucznym pochodzeniu obu księżyców Marsa.

Wykonane z małych odległości zdjęcia Czerwonej planety pozwoliły wyróżnić szczegóły wielkości jednego kilometra, a na niektórych obszarach nawet kilkudziesięciu metrów. Nie odkryto jednak kanałów marsjańskich, o których od stu lat dyskutowali uczeni. Występuje tam natomiast wiele dolin przypominających koryta wyschniętych rzek. Są również ślady przypominające ruchy lodowców. Jednak wody nie wykryto.

W warunkach chłodu i szczątkowej atmosfery marsjanskiej należałoby jej szukać — podpowiadają uczeni — ewentualnie pod lodowcami, głównie w okolicach nazwanych Hellas i Ellada. Ta ostatnia jest niecką o średnicy ponad półtora tysiąca kilometrów i głębokości czterech kilometrów, pokrytą grubą warstwą piasku I pyłów. Może jest to zamarznięte morze?

Kluczowym zadaniem jest badanie ewolucji zasobów wody na Marsie. Toczono także spory: z jakiego lodu — wodnego czy kwasu węglowego — powstały na Marsie czapy polarne? Dziś zgodzono się na kompromis — z jednego i z drugiego. Problem nie został jednak ostatecznie rozstrzygnięty... Niektóre szacunki wskazują, że ogólne zasoby wody na Marsie odpowiadają warstwie o grubości setek metrów pokrywającej całą powierzchnię planety. Automatyczne stacje kosmiczne, które wylądowały na Marsie, nie wykryły żadnych śladów życia, drobnoustrojów, cząsteczek organicznych czy roślin — traw, krzewów, drzew. Niewykluczone jednak, że w przeszłości, gdy występowały tam jeziora, szanse na powstanie życia były większe.

Właśnie aby odpowiedzieć na powyższe i inne pytania — a potrzeba tych odpowiedzi setki tysięcy, jeśli chcemy na serio myśleć o wylądowaniu człowieka na Marsie — należy realizować kolejne, coraz bardziej złożone ekspedycje. Takim właśnie ważnym krokiem na drodze do przygotowania pilotowanych wypraw na Marsa będzie projekt „Kolumb".

Istotą projektu „Kolumb" jest wprowadzenie aparatu kosmicznego na orbitę wokołmarsjańską, wysłanie na powierzchnię planety małego robota — marsochoda, wyposażonego w system czujników do badań sejsmicznych i w zestaw aparatury do analiz fizyko-chemicznych gruntu oraz umieszczenie w atmosferze dryfującej sondy.

Aerostat i marsochod — unikalne rozwiązania zaproponowane i realizowane przez specjalistów radzieckich — to nowy element badań tej planety, pdróżniający wyprawę radziecką od zaplanowanej mniej więcej na ten sam okres amerykańskiej misji „Mars-Observeur".

dokończenie na str 31



Jurij Zajcew



www.t2e.pl

Seria „Reduks" ma na celu przetworzenie – pełne zdigitalizowanie prasy komputerowej z początków okresu informatyzacji naszego kraju.

W odróżnieniu od plików w formacie djvu i cbr gdzie format papierowy jest tylko zeskanowany i połączony w jeden dokument, **reduks polega na ponownym przełamaniu dokumentu, wraz z ponowną obróbką graficzną - często z ich ponowną kreacją**. Tam gdzie to możliwe zachowana jest grafika z pierwowzoru, jednak jeśli materiał jest zbyt złej jakości - zostaje on podmieniony (oczywiście jeśli jest dostępny) na równoważny. Jeżeli uda się odnaleźć błąd w zamieszczonych listingach są one poprawiane tak, aby mogły zostać uruchomione na emulatorach lub mikrokomputerach.

Rozpocząłem ten cykl od „Bajtków", jednak mam nadzieję, że uda mi się wzbogacić kolekcję zdigitalizowanych i ponownie przełamanych periodyków także o inne ważne tytuły z lat ubiegłych, o których powoli zapominamy, a bez których współczesny obraz komputeryzacji wyglądałby zgoła inaczej.

Wydania „Reduksowe" są bezpłatne, nie roszczę sobie żadnych praw do zawartości przetwarzanych magazynów, jednak ilość czasu i pracy jaką wkładam aby mogły one trafić w Twoje ręce jest na tyle spora, że pozwolę sobie napomknąć iż możesz wesprzeć mnie przekazując dotację w dowolnej wysokości.

Jeśli posiadasz materiały fotograficzne lub edytorskie, które są lepszej jakości niż prezentowane w numerach, daj znać.



W ramach wolnego czasu odtwarzam stare magazyny komputerowe i prowadzę serwis emulacyjny try2emu. Jeżeli uważasz, że moja publiczna działalność możesz postawić mi kawę, a kawę lubię;)



https://buycoffee.to/t2e

Osoby, które wspierają serwis datkami* mają dostęp do reduksów w wysokiej rozdzielczości z usuniętymi reklamami.

W zależności od wysokości datku każdy otrzymuje czasowy dostęp do reduksów w pełnej rozdzielczości na miesiąc, 3 miesiące lub 6 miesięcy.
Osoby, które aktywnie wspierają projekt mają dostęp nieograniczony.